

## (1) HTML

### Syntax

< starting Tag > Hypertext < / ending >

#### (1) Text tag

→ ① Heading Tag - ⑥

< h<sub>1</sub> > AB < / h<sub>1</sub> >

< h<sub>2</sub> > AB < / h<sub>2</sub> >

< h<sub>3</sub> > AB < / h<sub>3</sub> >

< h<sub>4</sub> > AB < / h<sub>4</sub> >

< h<sub>5</sub> > AB < / h<sub>5</sub> >

< h<sub>6</sub> > AB < / h<sub>6</sub> >

#### ② Paragraph Tag -

< P > AB < / P >

#### ③ Pre Tag

< Pre > AB < / Pre >



## ② Structure Tag

- ① HTML tag `<html>`
- ② Head tag `<head>`  
`</head>`
- ③ Body tag `<body>`  
`</body>`  
`</html>`

## ③ Text formatting tag

- ① underline  $\rightarrow$  `<u> AB </u>`
- ② Italic  $\rightarrow$  `<i> AB </i>`
- ③ Bold tag  $\rightarrow$  `<strong> AB </strong>`

## ④ Meta tag

meta data

Data about data

`<title> Demo html </title>`

Extension of html file is written as .html

heading size decreases from  $h_1$  to  $h_6$

for SEO

Starting tag must have end tag

each tag must be in lowercase

title must be used in head tag

heading tags  $h_1$  to  $h_6$  must be used under body tag

Google has chrome for search engine



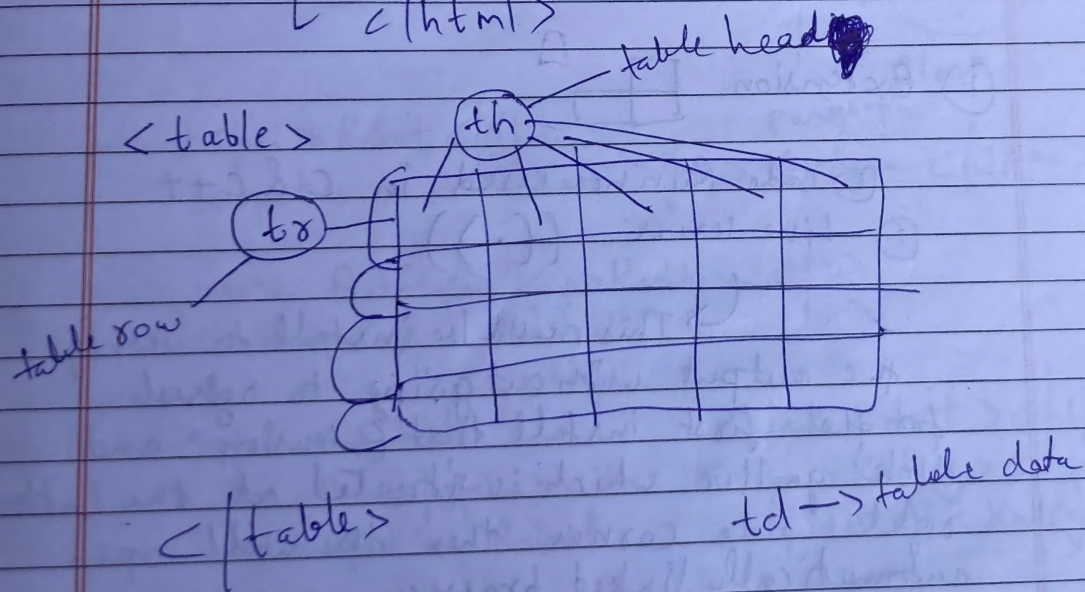
Meta data must be used under head tag

HTML file format

HTML file

```

<html>
  <head> <title> demo HTML </title>
</head>
  <body> <h1> AB </h1> to <h6>
</body>
</html>
  
```



```

<table>
  <tr>
    <th> Name </th>
    <th> father Name </th>
  </tr>
  <tr>
    <td> Azadwindu </td>
    <td> Mr. </td>
  </tr>
</table>
  
```



## ① Boiler Plate

complete HTML format

! → Ex net Observation

Visual Studio code

open desktop and create a new folder named anything to it and right click on it and select open with code that folder then click New file and write file name to anything with extension that HTML

## ① Extension

① Code Runner used in C & C++

② Live server ((.))

↳ This must be install to verify the output without going to refresh for that first install that extension and click go live which is situated at the bottom right side corner then this will open automatically linked browser

for go live to be proper work internet connection must be needed

for Boiler plate to be work must install before live server ((.)) extension then open that file name with - extension which you created then click shift + 1 you got it asks for Ex net Observation click that then you got complete HTML format called as Boiler plate

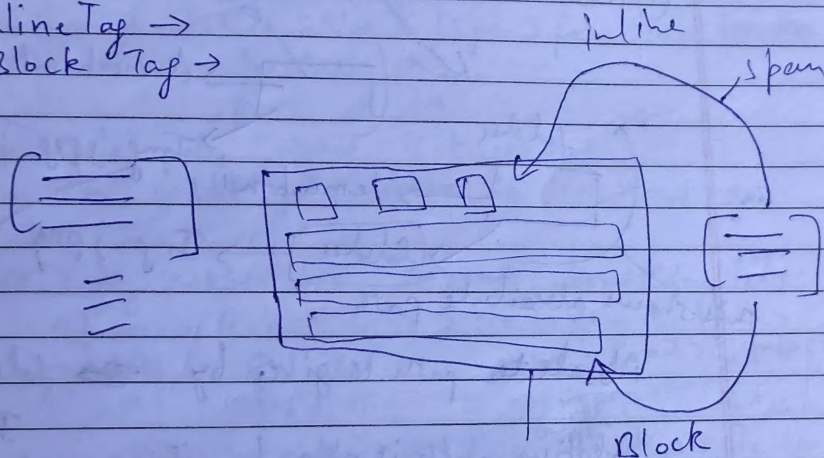


<!DOCTYPE html>

↳ this says about HTML5

(1) Inline Tag →

(2) Block Tag →



<span> Rohit </span> <span> Kadiyan </span>  
output is RohitKadiyan

<p> Rohit </p> <p> Kadiyan </p>  
output is Rohit  
Kadiyan

<span> Rohit  
</span>  
output is Rohit

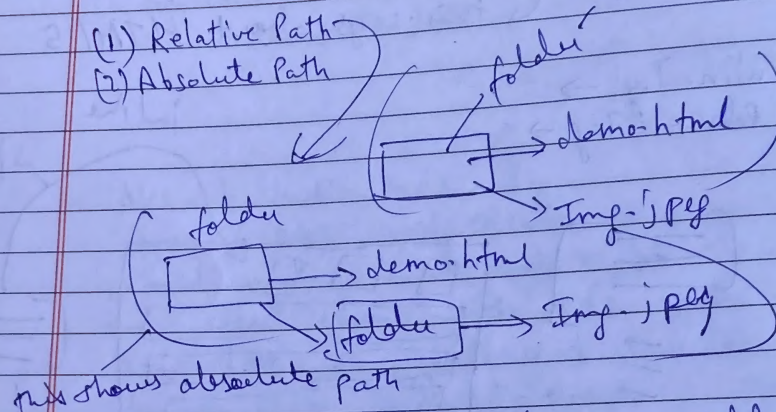
<span> Rohit > </span>  
<span> Kadiyan > </span>  
output is Rohit Kadiyan

<span>  
Rohit  
</span> <span> Kadiyan </span>  
output is RohitKadiyan

<span>  
Rohit  
</span> <span> Rohit  
</span>  
output is Rohit Rohit  
Not's means give space  
Not got RohitRohit



- (1) Relative Path  
(2) Absolute Path



Absolute path is given by ~~folder~~ folder / folder /

relative path is given by if you are in the folder where all files located then simply say img.jpeg

(1) <HR> → use for horizontal line  
tag

→ has no ending tag

(2) <BR> → use for Break  
Tag

has no ending

mostly use in table as like bordering Tag  
and bottom like upper

for left and right side bordering use  
shift plus ! acts as |



`<span> ^ SNo. ^ </span>` for left side and right side

`<hr>`  
`<span> <hr> SNo. ^ </span>` for Top ~~border~~ and bottom border

if you apply space between | |  
then you'll get space in output too

`<br>` tag used for break something like  
`<span> Rohit <br> Kadiyan </span>`  
 output is Rohit Kadiyan  
`<span> Rohit </span> <br> <span> Kadiyan </span>`  
 output is Rohit Kadiyan

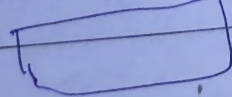
So, after break something text is to be written that comes in next line

`<input type=" " >` has no ending tag  
 tag input tag has fixed attribute here is called type  
 input tag has no ending tag

# How to apply attributes in HTML tag

`<input attribute1="value" / >` space attribute2  
`= "value" / space attribute3="value" >`

attribute defines what is this tag  
 what work in tags

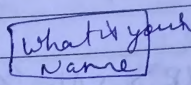
`<input type="text" >` value  
 output  a box you can write any text here



first attribute

second attribute

`<input type = "text" placeholder = "What is your Name"`

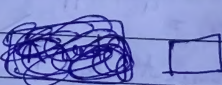
output is  on typing only your text shows

placeholder is 'just write something inside box for just showing

checkbox attribute for showing something like ☒ that's works on click on empty box as creating tick sign to that box

button attribute for showing something like  this has click option so this acts as a button

`<input type = "button" value = "Click Me" />`

output is 

for using checkbox apply type = "checkbox" instead of text

~~name = "checkbox"~~  
~~<input type = "checkbox" value = "Click Me" />~~

If you want to show the left and right sides as at proper place like right side seems border like so we pre tag similarly this works for left side too



`<th><hr><pre>| SNo.`

output is

`|</pre><hr>  
</th>`

other input tags are

`<input type="color">`

output is

filled color

default color is

black

when you click on filled color you can select any color

`<input type="date">`

output is

when you click on this box you can select date

`<input type="datetime-local">`

output is

when you click on this you can select date and you can see local time too



`<input type="email">`

output is

use this for writing email

`<input type="file">`

output is  No file chosen

use this to choose file

`<input type="hidden">`

output is ☐

you can click on that like button but when you give value by some text this works with button not with this

`<input type="image">`

output is

here we have image icon which to be valued not yet given any image so this is a icon

`<input type="month">`

output is

use this to select month



~~number~~

before typing

`<input type="number">`

output is



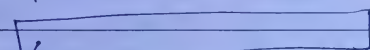
after typing



use this to write number

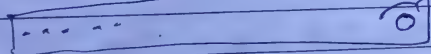
`<input type="password">`

output is

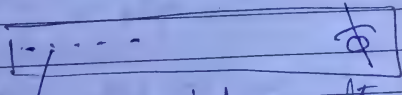


before typing

after typing



on click this



here is password shown after click

this



`<input type="radio">`

output is



Black filled color after you

click this



`<input type="range">`

output is

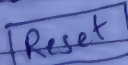


filled color

use this to increase or decrease some range

`<input type="reset">`

output is





use this to Reset

`<input type="search">`

output is

before writing

after writing

`<input type="submit">`

output is

use this to Submit anything

`<input type="time">`

output is

use this for select time

`<input type="week">`

output is

use this for select ~~the~~ ~~week~~ ~~of~~ ~~month~~  
weeks as per the month

# CSS → Cascading Style sheet

- ① Inline CSS
- ② Internal CSS
- ③ External CSS



`<starting tag style = "property: value;" >` `</ending tag>`  
 such as `border`, `color`  
`Green House` is a property value  
 Date `that` is a property value  
 Page No. `value`  
 attribute

Hypertext is text  
 to which whole CSS is  
 applied

`<h1 style = "border: 2px solid red;" > A red </h1>`  
 its width of border

Inline CSS means CSS is applied under any tag

`<h2 style = "color: green;" > This is the Heading </h2>`

Always style attribute is fixed to starting tag if you applying CSS, property and its value must be in lowercase

`<u style = "border: 2px red solid;" >underline tag </u>`  
`<u style = "color: green;" >underline tag </u>`  
`<i style = "border: 2px red solid;" >italic tag </i>`  
`<i style = "color: 2px red solid;" >italic tag </i>`  
`<strong style = "border: 2px red solid;" >bold tag </strong>`  
`<strong style = "color: green;" >bold tag </strong>`

output is

italic tag

underline tag  
 bold tag  
 black color  
 red color

underline tag

bold tag

italic tag

italic written



Other tags are `<ul>`, ~~`<ol>`~~, ~~`<col>`~~  
`<video>`, `<audio>`, `<select>`

`<ul>`

`<li>AB</li>`

`<li>AB</li>`

`<li>AB</li>`

`</ul>`

for unordered list

output is

- AB That's
- AB mean
- AB output with dot sign prefix

`<ol>`

`<li>AB</li>`

`<li>AB</li>`

`<li>AB</li>`

`</ol>`

for ordered list, output is

1. AB
2. AB
3. AB

`<video>` → for <sup>embed</sup> video in HTML website  
`<source src="ghi.mp4" type="video/mp4">`  
`</video>`

`<audio>` → for embed audio in HTML website  
`<source src="ghi.mp3" type="audio/mp3">`  
`</audio>`

`<select>`

`<option>AB</option>`

`<option>AB</option>`

`</select>`

required

is used to select <sup>required</sup> item from number of items

option are used to what options are available during selecting



Output 4

ABV

When you click on arrow icon  $\nabla$  you get AB and AB as options for selecting. For selecting you have to click on the item which you want to select.

div tag acts as a container tag. It contains number of tags, if you apply any property on div tag then you get that same property on all tags which contained under div tag.  
there are number of tags  
 $\langle \text{div} \rangle$   $\langle / \text{div} \rangle$

If you want to apply two properties with style then use single style keyword then write other property by given space.

$\langle \text{div style} = " \text{border: 2px red solid; color: red; border-radius: 20px; width: 50\%; height: 300px} " \rangle$

Here are number of tags such as  $\langle \text{h} \rangle$ ,  $\langle \text{input} \rangle$   
 $\langle / \text{div} \rangle$  all the property are contained under single quotes such as " "  $\rightarrow$  must use semicolon to each property at the end

width must be in percentage, height must be in pixel  
do not apply semicolon at the end of attribute

border-radius is another property to rounded the borders at corner that's must be in pixel

$\langle \text{a} \rangle$  tag anchor tag, if you write something in that's a file name.



That file has different number of page data so we linking that file name to other website page by given absolute path

for that we use keyword href

`<a href=" " >` A file, which you want to  
path link end tag  
here is file name `</a>` of anchor

if you are under folder admin, there's a filename called table.html

`<a href="table.html">Aad </a>`

finally href is used to link one file to other, one webpage to other

if in case you give href a path # that's, means you can't exit current webpage even still you trying to move to other page by clicking file name or you click file name but you got same current page

if you use div tag then there number of tags be shown inline or their output break `<br>` tag after each tag at the end

`<div>`

`<input type="date"> <br>`




`<input type="email"> <br>`

`</div>`



Selecting a color from already developed website

right click anywhere at the chrome browser to the desired website then click on inspect left section for HTML and right section for CSS

for selecting a color you have to choose CSS section then find out this syntax, color: #1c1e21; like that, then click on filled color box then click on  filled color this icon then you get , move this to desired color place and click on that in simultaneously, circle after  filled color

Now has desired color of desired place of desired website

that's comes under CSS section

Now, you got #1877f2 a desired color value of desired place

<button> tag  
<button> </button>

Internal CSS ~~has three tags~~  
~~<style>~~ ~~</style>~~  
~~<style>~~ ~~</style>~~  
~~<style>~~ ~~</style>~~



syntax

`<style>`  
`</style>`

Internal CSS has three types

- (1) Tag selector → It is used under tags if you apply on all that's apply on all div's
- (2) ID selector → for selecting Id put `id = " "`
- (3) class selector → for selecting class put `class = "anything class name here"` anything name here

ID be unique ~~for~~ so it can use once

But class also be used more than once

for Tag selector put `div` keyword to style tag that's must be under head tag for applying any property to that

```
<style>
  div {
    color: red;
  }
</style>
```

for Id selector put first # sign then Id name to style tag

```
<style>
  #head {
    color: red;
  }
</style>
```

for class selector put first . dot sign then classname

put under body tag



`<style>` / class name

Green House  
Date \_\_\_\_\_  
Page No. \_\_\_\_\_

• head {  
color: red  
}

`</style>`

`<head>`  
`<style>`

div { background-color:  
black;  
}

# win {  
color: blue;

h1 {  
color: blue;

• winner {

`</style>`

`</head>`  
`<body>` `<div>`

`<h1>` Azad `</h1>`

`<h2>` Azad `</h2>`

`<h3 id="win">` Azad `</h3>`

`<h3 class="winner">` Azad `</h3>`

`</div>`  
`</body>`

background properties

`<body>` / class selector  
`<div class="Azad">` Azad `</div>`  
`</body>`

`<style>`  
div { width: 100px;  
height: 300px;



border: 8px red solid;  
background-color: red;

this may be relative or absolute

→  
~~• A red~~  
background-image: url("file name");  
background-repeat: no-repeat;

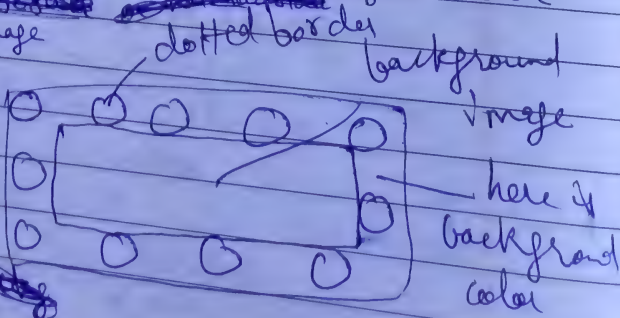
→  
we use div selector because we want to use container and class here to use background image automatically images ~~automatically~~  
in internal css div width and height as per the selector applied to it.  
background-repeat: no-repeat; that's means there is no repetition of one pic more than once on single webpage

background-size: 100% 100%;  
we use this to present background image width and height

here first 100% is for background image width and other is for height

background-clip: border-box;

div ✓  
before that we use this to put background image under border not on border  
we put change for image  
output is



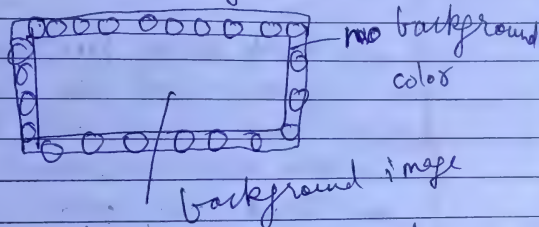
~~we use this to put~~



so that you can see background color

background-clip: ~~border-box~~ content-box;

we use this to put border outside div  
we use this to hide background color  
output is



generally in dotted border we get background color ~~that's why~~  
~~background-clip~~

background-position is used for background image position where to be located either at left right top and bottom

background-position: center;  
background-position: right;  
background-position: left;  
background-position: bottom;

background-origin: content-box;  
is same as background-clip: border-box;

background-origin: border-box;  
output is ~~background image starts here~~ border dotted one

The diagram shows a rectangle with a dotted border. The text 'background image starts here' is written inside the rectangle, and 'border dotted one' is written outside the rectangle to the right.



rgba stands for red green blue alpha

rgba (red of here, green percentage here,  
blue percentage here, alpha)

also, a rgb available

red color lies between 0-255

green color lies between 0-255

blue color lies between 0-255

alpha lies between 0 to 1 (here also  
considered 0.1, 0.2)

alpha keyword shows opacity or transparency  
of the color that's means more the  
opacity we can watch through that what  
is behind

If we try to lesser the transparency color  
goes solid

rgb(255, 0, 0) for red color

rgb(0, 255, 0) for green color

rgb(0, 0, 255) for blue color

If we try lesser the value from 255 to  
lower the color getting fades away on  
lowering the value

rgb(0, 0, 0) for black color

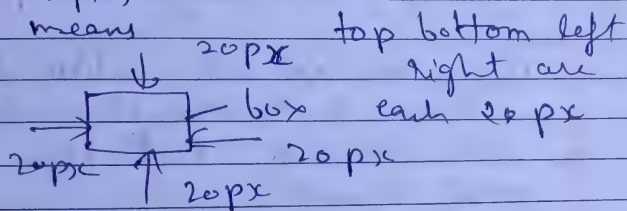


Margin is the space from the browser boundary to the div outside

padding is the space from div to inside (that may be box or other)

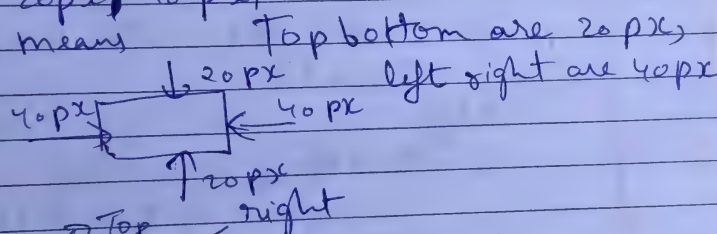
① Margin: 20px;

that's means



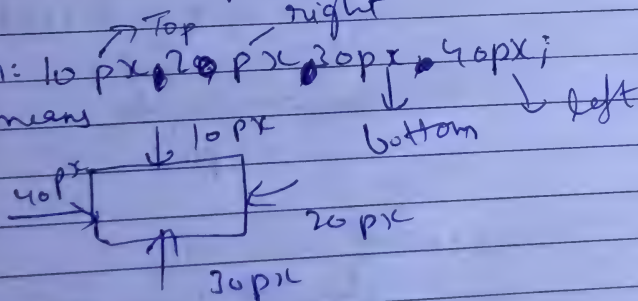
② Margin: 20px, 40px;

that's means



③ Margin: 10px, 20px, 30px, 40px;

That's means



Browser is actually is a parent element to the ~~other~~ other which is inside, so for browser that element which inside ~~browser~~ parent element that's cover 100% value and browser covers 100% value, if in case a other new element is inside that element which is already under parent then that element



is also 40% for approaching the  
parent element  
not

parent element <sup>^</sup> be 100% for responsive  
website <sup>we should give other element to the 100%</sup> and we don't touch that element  
width and height, if you want to do that  
you may use margin and padding

If in case we not consider browser as parent  
element to the other element then for  
a new element that other element acts as  
parent element

margin: auto; for center something

<head>

<style>

.Aradf

border: 2px red solid;

width: 40%;

height: 20px;

margin: 4px;

.winner

border: 2px green solid;

width: 100%;

height: 4px;

</style>

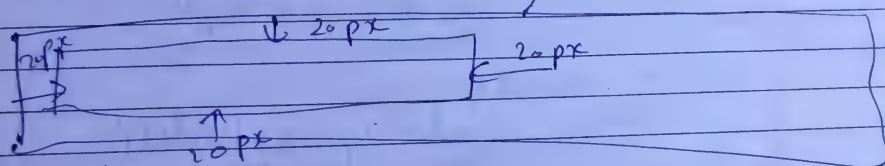
</head>



```
<body>
  <div class="winner">
    <div class="Arad"></div>
  </div>
```

we here takes winner class as parent class to the Arad class

output is



```
<head>
<style>
```

```
.Arad {
  border: 2px red solid;
  width: 400px;
  height: 20px;
  margin: 4px;
```

```
}
.winner {
  border: 2px green solid;
  width: 400px;
  height: 50px;
```

```
-div1 {
  border: 2px green solid;
  width: 400px;
  height: 50px;
```

```
}
```



• div2 {

border: 2px green solid;

width: 40%;

height: 50px;

}

• div3 {

border: 2px red solid;

width: 40%;

height: 20px;

margin: 20px 40px;

• div4 {

border: 2px red solid;

width: 40%;

height: 20px;

margin: 10px 20px 30px 40px;

}

• div5 {

border: 2px green solid;

width: 40%;

height: 50px;

}

• div6 {

border: 2px red solid;

width: 40%;

height: 20px;

margin: auto;

}

</style>

</head>

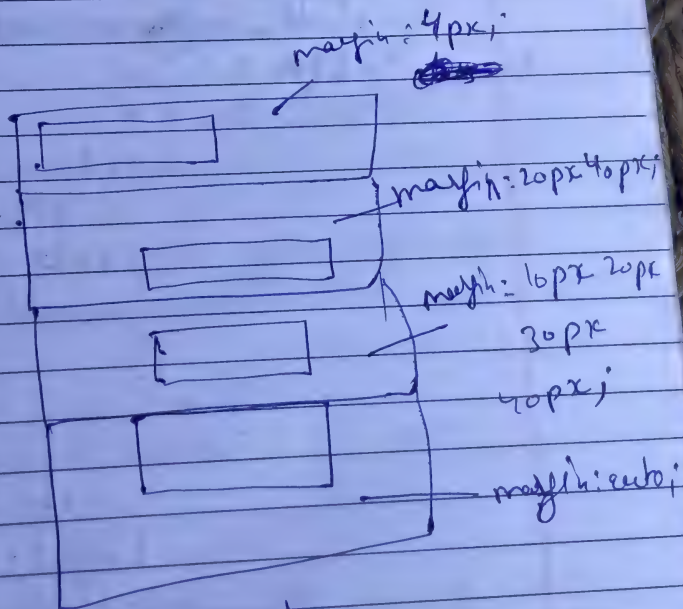


```

<body>
  <div class="winner">
    <div class="Azed"></div>
  </div>
  <div class="div1">
    <div class="div3"></div>
  </div>
  <div class="div2">
    <div class="div4"></div>
  </div>
  <div class="div5">
    <div class="div6"></div>
  </div>
</body>

```

output is



margin-top: 20px inside  
push element div from top towards  
bottom by 20px



margin-bottom: 20px;  
push element inside div from  
bottom towards top by 20px;

margin-right: 20px;  
push element inside div from right  
towards left by 20px;

margin-left: 20px;  
push element inside div from left  
towards right by 20px;

~~margin~~ overflow: hidden;

~~<head>~~ ~~<body>~~ ~~</body>~~ ~~</html>~~  
~~<style>~~

• Arad <

border: 2px red solid;  
width: 1000px;  
height: 20px;  
margin: 4px;

• Winna <

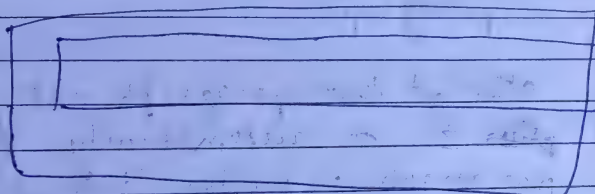
border: 2px green solid;  
width: 4000px;  
height: 50px;  
overflow: hidden;

</style>  
</head>



```
<body>
  <div class="winner">
    <div class="A red"> </div>
  </div>
</body>
```

output is



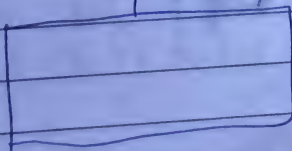
that's means class A red, hided by class winner so we use overflow hidden; property in parent element

Padding

padding-top; its push top side by any value or its expands towards top

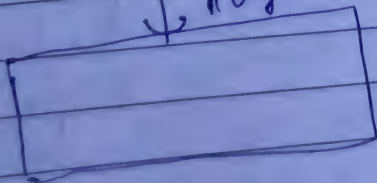
↑ padding-top: 20px;

output is



margin-top; its push top side by any value towards bottom but not impact on other sides

↓ margin-top: 20px;





font-size: 12px;  
which is used to increase or decrease the font size, this is applied to the div which contains that font

lorem is a inbuilt html and css paragraph for use. type lorem and then enter and you get a paragraph

alt after got that paragraph just ~~ctrl~~ type ~~ctrl~~ <sup>alt</sup> z to automatically adjust all paragraph as per the lines

Control z for backspace

Control s for save all the things what you write

Alt tab for easily switch from current tab to website or website to tab

Control ~~alt~~ P for comment  
for use select that line use this control P

Display ~~flex~~ properties:

① Display: none;

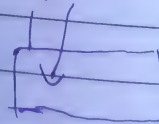
↓  
Properties value

② Display: show;

③ Display: flex;

background-color: red;

div





① Display: none;  
not hides only background but  
also the div

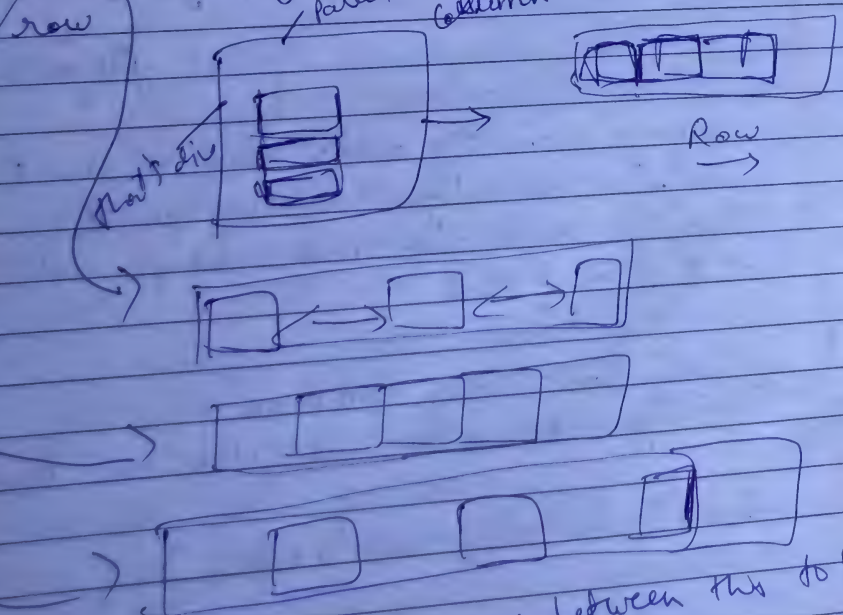
② Display: show;  
if you applied already display  
: none; display: show that  
content gain shows

③ Display: flex; always use on parent

- ① Justify - content: center;
- ② Justify - content: space-between;
- ③ Justify - content: space-around;

These three properties can't work without  
display: flex

Display: flex; convert column to  
parent column

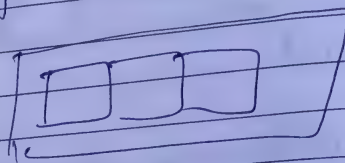


if actual is 10px after space-between this do be 10px  
so we have to take 10px as 9px



```
<div> box1  
      box2  
      box3 </div>
```

output



`text-align: center;` to put the text at the center

`border: none;` to remove the border

`<a class="any class name" href="filename.html"`

↓  
file path → </a>

always elements under a div must be of 100% at primarily then use padding or margin then you may change width of that if need that




`<body class="any class name">`

you can also use class on body tag

- ① hover → `box: hover { }`
- ② transition
- ③ transform



Three properties of cursor

cursor: pointer; output is   
 cursor: zoom-in; output is   
 cursor: zoom-out; output is 

transition can't work without hover

transition is to control the speed of the .box  
 hover

we use linear keyword to say anything going  
 with same speed

transform is to rotate the box

↓  
 property                      ↓  
                                     its value

Transform is of three types <sup>such as 120 deg</sup> degree enter here

① rotate ~~for~~ <sup>degree enter here</sup>

② ~~skew~~ skew <sup>such as 120</sup>

③ scale (1) that's same  
 as 0px 4px before and after <sup>this is laptop type</sup>

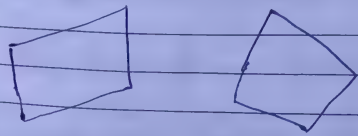
speed in seconds

scale (2) that's doubled the values  
 such as 0px <sup>width</sup> 8px height

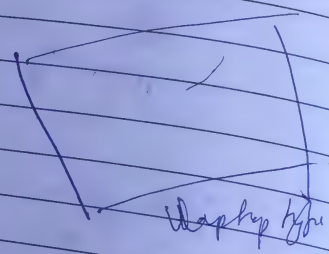
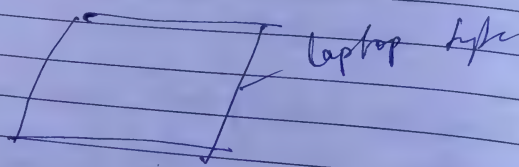
transition: all 10s linear;  
 ↓ that's applied in ~~the~~ same proportion  
 on whole



rotate transform outputs are



skew transform outputs are



transform: scale (2);  
that's doubled the values

we have to use transform under hover  
such as .box: hover {

transform: scale you can do on }  
.box { }

hover it to apply properties  
transition, cursor such as

transform: rotate (120deg);  
transform: rotate (90deg);

→ this used for if  
~~the~~ horizontal line  
to you have to  
rotate  
for this, use transform

shows like this  
(1)



rotate is to slight rotate the things  
~~show it to rotate~~

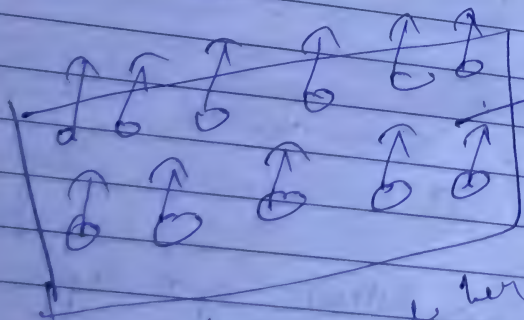
for using box - hover

we first create box { with border  
width & height

for hovering give background color to box.  
hover

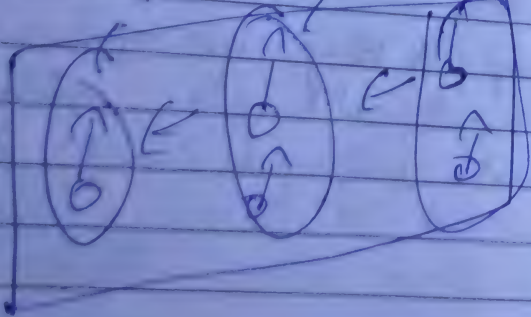
At primarily don't use padding  
use under div must be of 100% width  
not to reduce that

linear ~~and~~



color proportion  
changes as  
same in wheels

case  
word her hand her first her



color proportion  
changes in  
steps

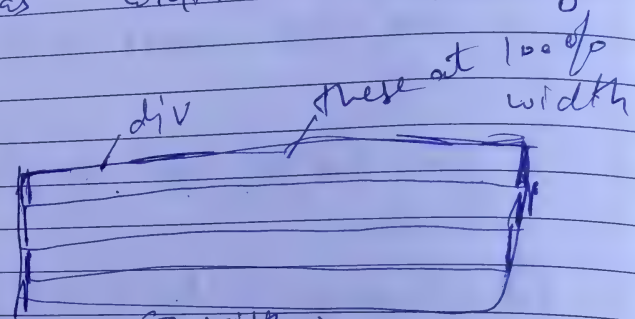


padding is background box

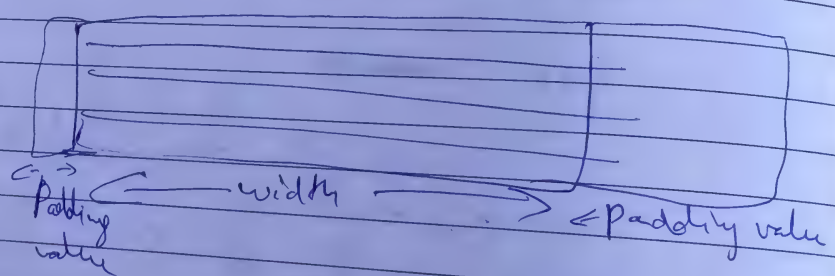
padding is width of the box ~~at~~ div  
to whom you applied padding and  
padding value itself

such as width value + padding value

output



on applying padding background box  
created



padding: 0px 12px for left and right

for proper fitting we apply padding not  
on input div its applied on outer  
div who contained input div



for responsive website ~~and~~ we use padding after width of inputs to be 100% then we do work on inputs div then apply padding on outer div which contained inputs div

class selector which is something absolute path to specific class

such as

```
<div class="box1" id="A2ad"
  <a href="#" > </a>
</div>
```

not to class of that tag

for selecting anchor tag or <a> tag use .box1 a {

```
<div class="anchor"> <a href="#" > login
  </a>
</div>
```

for selecting anchor tag or <a> tag use .anchor a {

text-decoration: none; to remove underline from anchor tag link such as forgotten password

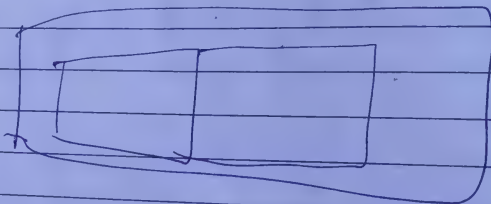
snipping tool is used to take screenshot of the selected area from any website



mostly uses for logo  
for use this search shipping tool in search  
bar then click windows icon + shift  
+ S then draw or outlined a  
specific logo or image to whom you want  
to save after selected then click ...  
three dotted line and click save

for easy movement of the input tag use div or  
that

padding works same as margin



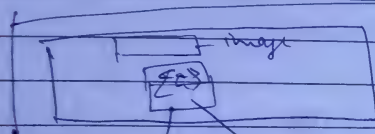
we want to shift left box to right  
side and right box to right side.  
so we have option of margin-left for  
left box and margin-right for right box  
in opposite we may use padding  
can do both works such as ~~padding~~ margin-left  
and margin-right for both the boxes



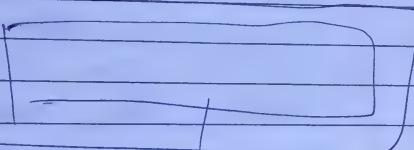
How is to use cursor: pointer property on the website or its related website ka

1. Sabse pehle design or structure tayyar Kige jata hai fir write Kige jata hai text ko

such as 100% width div or box



600 x 1500% width



600 x 100% width

we try to use div less so we may use twice or more anchor tag under a single div but note that we do use color properly on the anchor tag so that we may use different div for each anchor tag

after create structure then we write text or elements because if we write text before structure created that's increase the height of the box where written

Kindly note that we use div as a mail tag to be come under display: flex; property for converting column to row

we have to create whole structure whole structure not partial one

should



If you want to give margin from left to right you may use margin: auto to or (or say both) center this

सबसे पहले नीचे वाली है दिव लगाने फिर नीचे बहरी है बोर्डर लगाने फिर इन नीचे के गुदा जो भी काम करना है वह करना है चाहे photo लगाने है

display: flex; works on even style class as or column to convert

<div> to row  
<input>  
</div>

So we

if you apply width: 300% as that's width already given to the button tag

With the button tag on short the page the button icon goes out from our layout so we apply that width to the div which applied on that button tag and in that case button must be of 100% width

if you take div <div> then you have to take another div on that now inner div must be of 100%



and enter div of as you need  
even on anchor tag text as

```
<a class="anchor">find talent</a>
```

so you have to div on that  
take another  
class= anchor512

```
<div><a class="anchor">find talent</a>  
</div>
```

now anchor class width is 100%

and you now set the width accordingly  
to anchor512 as width: 4px;

display: flex; not applied on another div it's  
applied on real div as  
class="anotherdiv"

```
<div><div class="real"></div></div>
```

you applied on real class div as you need

padding

if

Search

after 4px left padding output is

Search

texting start from here



/607

Simple

after padding - left: 4px

Simple

class="button"  
<div> <button class="button"> Sign Up </button>  
you have to apply background color on button not on another div </div>

height is applied on real div in all the cases not on another div

<div class="anotherdiv"> ^ <div class="button">  
you have to apply height on button class div not on another div class



in the case of anchor tag  $\langle \text{div class} = \text{"anchor"} \rangle$   
 $\langle \text{div class} = \text{"another div"} \rangle \langle \text{pdiv} \rangle$

~~again have to apply font-size~~  
~~padding-left: 8px; padding-right: 8px;~~  
~~padding: 8px;~~

With padding-left: 8px; width increases  
 so you have to sure reduce width  
 according to need too

~~padding-right: 8px;~~

With padding-top: 8px; ~~width~~ height increases  
 so you have to sure reduce ~~width~~ height  
 to need too

after padding-top: 8px;

Search

before

Search

Ctrl plus F for search any keyword  
 across the project

if you use <sup>pre under</sup> paragraph tag then you have







```
100% { background-color: red; }  
}
```

```
<style>  
-box {  
  animation: my all linear 1 1;  
}</style>
```

Same speed of whole once  
↓  
animation

twice  
↓  
animation

```
<div class="box">  
  content  
</div>
```

0% → 100%

for apply animation

first add this

```
<style>  
-box {  
  animation: my all linear 1;  
}
```

anything name here  
once animation

```
@keyframes {  
  0% { background-color: blue; }  
  50% { background-color: green; }  
  100% { background-color: red; }  
}
```

anything name here



animation not applied on image its  
applied on button or on .box

<style>

button { animation: my 2s linear 4; }

@keyframes my {

</style>

0% border: 1px solid red;  
background-color: blue;

It delay time  
mean animation  
stop after how many  
seconds on  
after  
restart

### Google font

Search google font on google search engine  
and you have got website similar to this  
<https://fonts.google.com> ~~font~~ then choose whether  
you need to use fonts or icons, so here  
we are selecting and searching fonts so click  
on that and now click font families such as  
Roboto, Poppins, even you can search font  
family in search bar such as (Search fonts) this  
Now scroll down and select font type whether  
its thin or italic by clicking this select thin  
100 (+) and this get added to your this  
option ☐ now click this ☐ and get a  
code to embed this to your ~~html~~ html  
file as under head tag ~~such as~~ such as  
<link ~~href~~ href="https://fonts.googleapis.  
com/css2?family=Poppins:  
wght@100&display=swap" rel="stylesheet">

under  
styles  
tab



then add this to your css file under style tag

```
.box {
  font-family: Poppins, sans-serif;
}
```

```
</body>
<h1 class="box">Arad</h1>
</body>
```

② font awesome <sup>use</sup> for use icon before and after text that's be under body under tags or before and after hypertext

```
</body>
<h1 class="box"> fontawesome, space (Arad) </h1>
</body>
```

of type which you apply

such as google icon  $\leftarrow$  `<i class="fa-brands fa-google"></i>`

other thing is use cdn of font-awesome for working of whole font-awesome that's used under ~~font~~ head tag such as ~~link~~ <sup>for</sup> (or abbreviation is `</>` for copy and paste to your file for cdn of font-awesome, search cdn of font-awesome and you get [cdnjs.com/libraries/font-awesome](https://cdnjs.com/libraries/font-awesome) and do copy and paste

### ③ External CSS

Inline CSS	Internal CSS
style tag	style tag
as	as tag used
html built used	under head tag

External CSS  
has no style tag  
we have ~~create~~ <sup>create</sup>  
new file with .css  
extension for adding  
css rather than under style  
under head tag



CDN is content delivery network  
http is hypertext transfer protocol

External CSS

file(1) - external.html  
file(2) - external.css

if external CSS has no head of style tag under head tag

after that we link the .css file to .html file  
so we used this

add this under your head tag

< link rel = "stylesheet" href = "anything" file name here with .css extension

and .css file

border: 2px solid red; border-radius: 40px;

that's standard

and under body box

<div> </div>

And

if you take percentage under @keyframes you should be take as under sequence such as

@keyframes my {

0% {

100% {

200% {

300% {

up to

1000% {

}



for using images under animation use

@keyframes my {

0% { background-image =

url("xpd.png");

background-size =

100% 100%;

<body>

<div class="box"> </div>

for width  
for height

</body>

<style>

</head>

. box {

animation: my 2s linear 1;

border: 2px red solid;

height: 200px;

width: 300px;

}

that's necessary to use after

background-image ~~background~~

in keyframes under percentages  
such as under 0%.

@keyframes my {

0% { background-image = url("xpd.png");

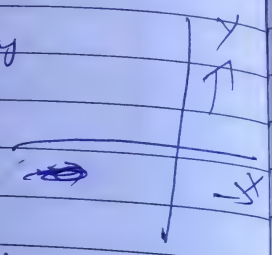
background-size: 100% 100%;

}



Bootstrap 4 → 5

- ① cdn
- ② start template
- ③  $l\text{-}sm-1 \rightarrow 8.3\%$  width  $\langle /div \rangle \langle /div \rangle$   
 $2 \rightarrow 16.6\%$  width
- ④ border class = "col-sm-1"
- ⑤ container  $(90\%)$  II<sup>nd</sup> container-fluid  $(100\%)$
- ⑥ mx-auto
- ⑦ mt-12 mb-12 ms-12 me-12
- ⑧ colors text-success, primary, danger, light and dark
- ⑨ Background color → bg-success, primary
- ⑩ m-4 apply on all sides



bootstrap is framework of CSS

already made class that's been used by some shortcuts or keyword to direct access that class not to create that one

currently bootstrap 5 is going on

for using bootstrap in your project we do use starter template rather than boiler plate because starter template is responsive

for obtaining bootstrap starter plate search introduction of bootstrap and you get page like <https://getbootstrap.com/docs/5.0/getting-started/introduction/>, now scroll down and find starter template then copy and paste in your project, as we know link tag is already



included in this ~~bootstrap~~ starter plate, so we need to add again link tag in the head tag don't

actually bootstrap works in html file extension so we to create anything file name with .html extension, now remove unnecessary files which you don't need such as script src

bootstrap also uses as class under body tag but don't need to create something like a box ~~under~~ CSS

for here we use just keywords like border, bg for background, we to use these keywords under class under body tag

for multiple properties class to look like this

```
<div class="border m-4 mt-12">  
    </div>
```

but something is that we to use class under div

for searching CSS terms in your bootstrap website use Introduction of bootstrap

webpage or [getbootstrap.com/docs/5.0/getting-started/](https://getbootstrap.com/docs/5.0/getting-started/introduction) then search for introduction

search class bar and enter CSS property which you want to use



col is for column

col-sm-1 is standard with width 8.33%  
2 width 16.66%

border is used for border property

container is used for div with 90% width

container-fluid is used for div with 100% width

mx-auto is used for margin auto across horizontal or x-axis

my-auto is used for margin auto across vertical or y-axis

mt-12 used for margin-top: 12px;

mb-12 used for margin-bottom: 12px;

me-12 used for margin-end: 12px;

ms-12 used for margin-start / margin-left: 12px; / margin-right: 12px;

color → text-success for ~~blue~~ green color text

text-primary for blue color text

text-danger for red color text

text-light for white color text

text-dark for black color text

bg-success for green color background with

bg-primary for blue color background with

bg-danger for background with red color

m-4 used for margin: 4px; applied from all the sides

p-4 used for padding: 4px; applied from all the sides  
height property is manually change there's no key word related to that in bootstrap

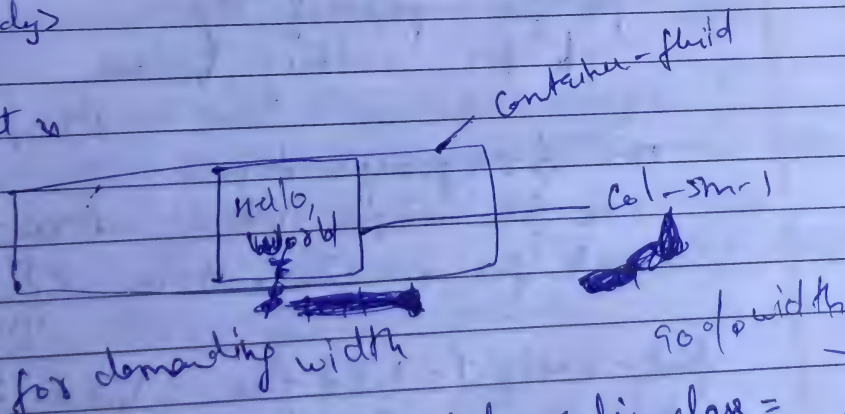


pt-12 used for padding-top: 12px;  
pb-12 used for padding-bottom: 12px;  
pl-12 used for padding-left: 12px;  
ps-12 used for padding-start / padding-left: 12px;

mt-5 has final limit is 5  
so its can't work on mt-6

```
<body>
  <div class="container-fluid border bg-primary">
    <div class="col-sm-1 border bg-danger mx-
      auto"> <h1 class="text-danger bg-light mt-2">
        Hello, world! </h1> </div> </div>
  </body>
```

output is



With bootstrap if you take `<div class="col-sm-1"> </div>` and put that  
so you have to take div on that `<div class="container">`  
`<div> <div class="col-sm-1"> </div>`  
`class="container"` `</div>`

~~If you want to take height 100% or so~~



~~set the font for height too~~  
~~for height: 50pt~~

~~generally height with 50pt or width with 50pt and similarly for top and bottom margin~~  
~~width and height~~

~~for input width too~~ In bootstrap there is  
~~add w too~~ no width and height property

col-sm-1 goes upto 14 such as col-sm-14  
 margin of all sides top, bottom, left and right goes upto 5 level, similarly level 5 to padding

pl for padding left

pt for padding top

pb for padding bottom

px - for both sides left

and right

py - for both sides

top and bottom

mt for margin top

ml for margin left

mb for margin bottom

mx - for both sides left and right

my - for both sides top and bottom

font-weight - bold for

btn-primary for button color blue

text-center for locating the button at center and text such as headings at center

d-flex for display: flex;



btn - success for button color green

~~display-1~~ works as ~~h1~~ font size  
~~display-2~~ works as ~~h2~~ font size  
~~display-3~~ works as ~~h3~~ font size  
~~display-4~~ works as ~~h4~~ font size  
~~display-5~~ works as ~~h5~~ font size

if want to increase col-sm-1 then you have to take div on that to increase that col-sm-1 upto 14 such as col-sm-1 without that that's not work

~~height in percentage not generally works so you have to take height in pixels even if you want to take 100% of the page~~

~~if you want to take no margin from body to div then you have to take container-fluid instead of container~~  
~~in <div class="container"> <div class="col-sm-1">~~

if you have anchor tags so you have to div take like this

Col 1   
 `<div class="container">`  
 `<div class="col-sm-1">` `</div>` `</div>`



for 100% width use container-fluid  
in div class  
for 90% width use container m div  
class

for demanding width use col-sm-1 upto  
24

| col-sm-1 | width % |
|----------|---------|
| 2        | 16.6 %  |
| 3        | 24.9 %  |
| 4        | 33.2 %  |
| 5        | 41.5 %  |
| 6        | 49.5 %  |
| 7        | 58.1 %  |
| 8        | 66.7 %  |
| 9        | 74.7 %  |
| 10       | 83 %    |
| 11       | 91.2 %  |
| 12       | 99.6 %  |
| 13       | 107.9 % |
| 14       | 116.2 % |
| 0        |         |

if in case you are in the third div

<div >

<div >

<div class="div3">

</div>

next's

you are working under  
last div so don't touch other div's

</div>

</div>



in bootstrap we prioritize to use less div even still the webpage is automatically responsive

previously was different

But we do take div & then do other div so that such as

~~other thing be changeable and seal~~  
div has width of 100% to be sure

But now that's not need to such as on all properties majorly such as button input heading paragraph and also we can't have too div's on that

real  
> (div) (div) </div> </div>

for why display: flex;   
 ~~property~~ must be under div's   
 each tags

such as <hr> tag, <p> </p> </div>  
tag and may more   
 such as <div> <hr> </hr> </div>  
<div> <p> </p> </div>  
<div> <input> </input> </div>



`max-auto` for center the box not applied to text

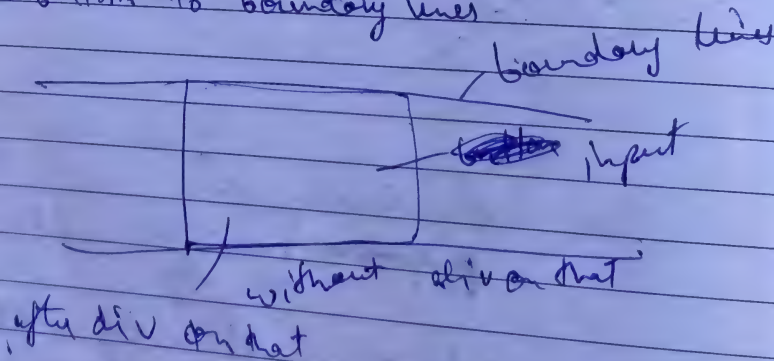
for text's use `text-align`

in bootstrap you can use margin sides even you applied `margin-auto`, here it is

such as  $\left( \begin{matrix} \text{max-auto} \\ \text{mt-5} \end{matrix} \right)$

but previously in CSS that was not working

Sometimes we takes `div` in a optional way its reduces the height of the components to 50% otherwise components touches both top and bottom to boundary lines





bootstrap

Green House  
Date \_\_\_\_\_  
Page No. \_\_\_\_\_

In manual scale is important to structure the website

such as take a 100% line either horizontally or vertically

→ 100%

now put the relevant components on that

40% 20% 40%  
| |  
div 100% →

now setting the components percentage on that  
and you may adjust the percentages according to the need

40% 18% 41%  
| |  
div 100% →

same goes with vertically

100%  
↓

100  
82  
18  
41  
41  
82



initially a div automatically takes some padding, you have to put padding  $p=0$  for no space <sup>so</sup> from outer container to inner container

In bootstrap we to use less div and classes

bg-secondary used for light black color

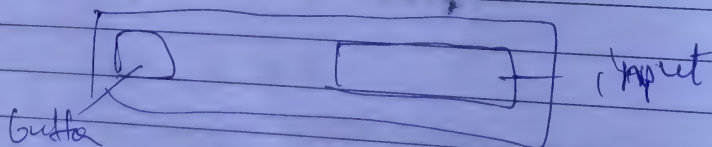
bg-warning used for yellow color

bg-info used for light blue color

bg-white used for white color

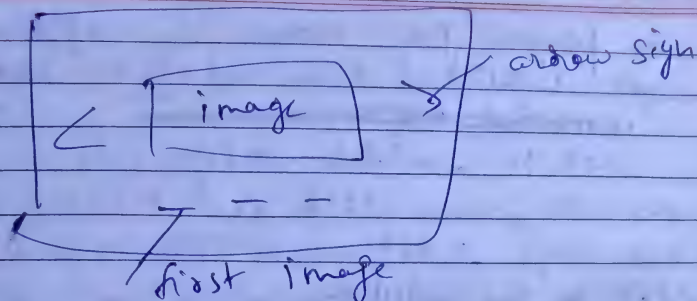
(1) Navbar commonly known as header but it's of its type

Navbar has button input ~~as~~ as per need you can add



(2) ~~Carousel~~ Carousel its like slide show number of images but it has option for you can shift from one pic to other by arrow signs < >






(3) Snippet is already made code for specific purpose such as login form but can't use them directly for the reason of you duplicacy so you have to apply changes to that such as change button text and color and background color as per your need

for searching snippet of code just type snippet of login form in bootstaps means login form here purpose In either case bootstaps code can't run into bootstaps, so you have to search according to that.

after get code copy that and paste your project if your project of html that's be pasted under <body> tag now run that

as you want to change text of button as sign in just right click that and click on inspect now click on this icon  on click that's must be turn blue after that click



such as how Sign button  
on page for change in after click you  
get highlighted line and click ~~that~~ for  
~~select that whole line~~ now ... (three dots)  
which is available at left side or start of  
the code, now click that three dots and  
now click edit as HTML and now click  
ctrl A for select the whole line and ctrl C  
for copy that, now move to your project  
now click ctrl F for search something

This way fast your work rather than searching  
whole project

In meanwhile button can't work without  
JavaScript that is a mechanism which handles  
everything behind the bootstrap

Bootstrap 3

Bootstrap 4

Bootstrap 5

Container size 1170px

1140px

1320px

column's available in sizes  
are sm, md, lg, xl

column's available in  
sizes are sm, md, col,  
lg, xl

column's available  
in sizes are sm,  
md, lg, xl, xxl

Non-responsive layout

Responsive  
layout

Responsive  
layout

Bootstrap is the most  
stable version of  
bootstrap

Bootstrap 4 is a  
newer version of  
Bootstrap with new  
components and responsive  
layout

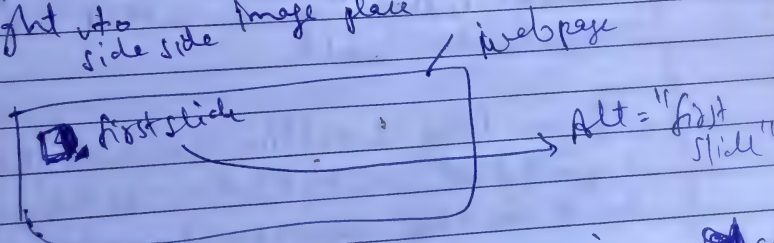
Bootstrap 5 has  
smooth interface  
here query  
replaced by a  
JavaScript



↳ script tag with src of cdn or code of jquery or jquery is the reason of working of buttons in bootstrap

↳ js here stand for javascript  
if that script in your project the button may be seen but not working at all

Alt attribute under image that's a specifying text showing image of which type if your internet is slow or not working image may be not loaded in that case you get that written text on your web page just right into side side image place



In bootstrap 4 script uses are `code.jquery.com`, `cdn.jquery`, `jquery` (maybe 2 or more)

In bootstrap 5 `cdn of jquery` is necessary to use (maybe 3 or more)

That's be used under body tag  
in prebuilt starter template has already these scripts to bootstrap version such as bootstrap 4 or bootstrap 5 according



if you applied margin-bottom this may increase height so you have to use height manually by CSS as per need such as

`<head>`  
`height { height: 200px; }`

`</head>`

In bootstrap if first applied padding such as p-0, if need, then remove the border such as border: none then now apply margins

such as border margin-top, bottom ~~margin~~

then again apply border border-primary for clarity is that your content or components is in under border

If you doing scrolling down the page and you don't want that header to be scrolling down too then use position: fixed; properly

if you use font-awesome icons that's not works or sometimes not visible then convert regular to solid such as in `<i class="fa-light">` to `<i class="fa-solid">` `</i>`  
 same for regular one

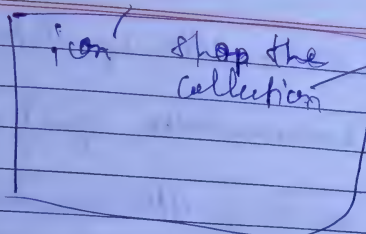
border goes from 1 to 5

Another property if you have situation like



label

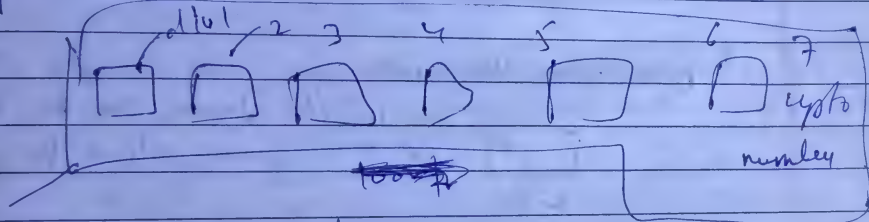
Green House  
Date \_\_\_\_\_  
Page No. \_\_\_\_\_



it should be  
start behind the  
shop then  
use span tag  
on collection and  
give that ~~label~~  
name's margin

```
<p class=""> Shop The <br>  
<span class="margin">collection </span> </p>
```

for responsive website these are clarity's

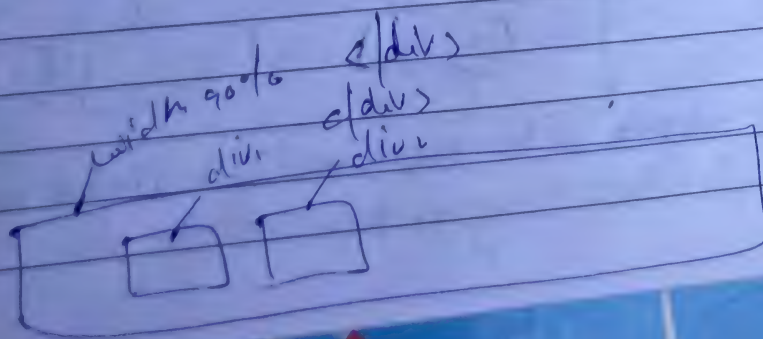


this width covers 90%

if you want take less div you may use  
col-sm-3 such as, col-sm-4 for width-90%

if you want to take number of columns as in  
single row you may use class="row"> <div>

such as <div class="row">  
<div class="border col-sm-3  
col-lg-6">






col-sm for devices width equal to 768 px



col-sm for devices width equal to 576 px

col-lg for devices width equal to 992 px

col-xl for devices width equal to 1200 px <sup>as same</sup>

~~for text not come, get from div, button~~  
~~for text not come, get from div, button~~  
we use some tools such as  
as go on inspect and click this I can 

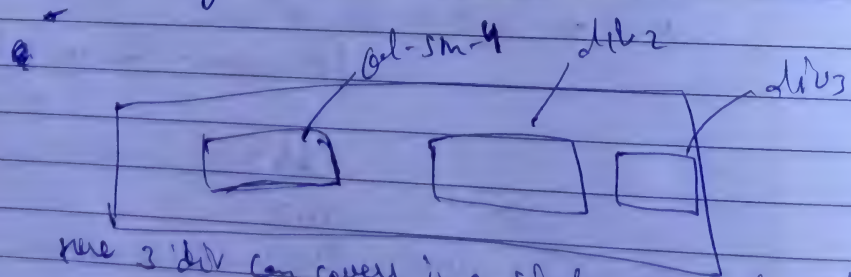
Dimensions: Responsive for just sample checking  
for width from sm to lg, x/

 x  that's to be put same

change this according to need like on  
992 px width, col-lg applied to webpage

here, col-lg-6 means large columns but  
that's also take width of the small one such as  
col-sm

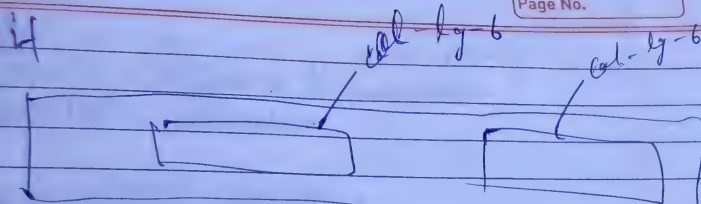
col-sm-6 for 49.50% width that's, also applied  
on col-lg-6 ~~but that's not means~~



here 3 div can covers in a single row of width  
of 90%

previous  
font-size  
we don't change  
font size  
or may  
more





Here 2 div can covers in a single row of width of 90%.

if you applied both on same div that means div has both properties such as

`<div class="row">`

`<div class="col-sm-4 col-lg-6">`

`<div class="col-sm-4 col-lg-6">`

`<div class="col-sm-4 col-lg-6">`

That means on width 576 px, this takes

2 div across a row

but simultaneously if you increase width,

to 992 px ~~by increasing the width~~ 2 divs

can cover by a single row and other div in a

That's a responsive, basically next row  
but increases or decreases randomly height of row

That means at large column stay at width

99.5% such as `col-lg-6` for 992 px width

You have to design start from `col-sm` at inspect  
pixels width 576 then set the `col-lg`  
which is at pixels width 992

~~you have an alternative way to display flex~~  
~~or to fix it~~ `<div class="row">`



~~at 576 px at 992 px~~  
~~col-sm-4 col-lg-6~~  
~~col-sm-2 col-lg-6~~  
~~col-sm-2 col-lg-6~~

~~you don't need to~~  
~~padding about 8px at padding~~  
~~as per~~

we to use this on button with the support of  
 div  
 such as  
`<div class=" " > <button> </button>`  
`</div>`  
 col-sm-4 col-lg-6 for text is contained  
 under border at both pixels 576 px or 992 px  
 button

### Javascript

A website where functionalities not working  
 such as onclick button not submitting or either  
 working too that's called static website

we turn this static website to dynamic web site  
 by adding functionalities after added functionalities  
 that's called dynamic website

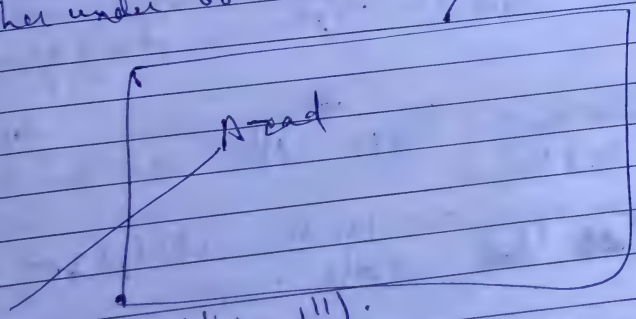
Javascript is easy, fast, here everything is  
 object such as h1, h2 tags here are objects



client side ~~is~~ for just viewing that website not to change that

server side for just viewing and change the website functionalities too

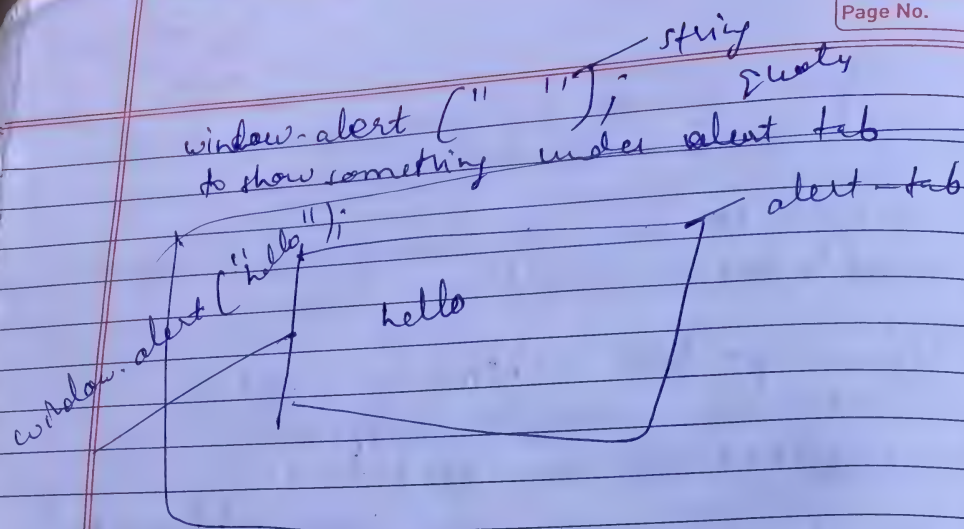
Javascript uses in html by script tag such as `<script>`/`</script>` that's can be added either in head tag or in body tag  
`document.write(" " );` string notations or double ~~quotes~~ quotes  
 uses to write something under body or either under browser



`document.write("Azed");`  
 here Alphabet is a character, a group of character is a string  
 A character can be noted by single quotes such as `document.write('A');`

→ This is either a alphabet and a character too  
 That's not means single quotes can't work on string and same for character as applying double quotes on it





without window keyword this works too such as `alert(" ");` string quotes

`console.log(" ");` string quotes do  
and find error this use to debugging  
if you apply this under `JavaScript` tab this applied on whole script

for checking you applied `console.log(" ");`  
go on inspect and click on console and you get console window

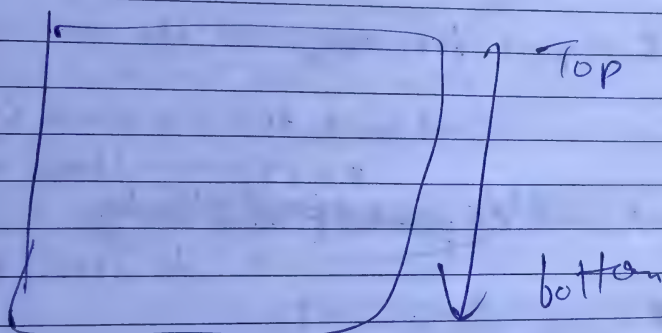
Operator is to perform something  
operator operates on operand  
such as  $c = A * B$   
here  $c$  is operand  
operator



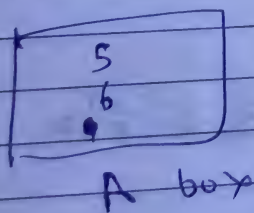
unary operators such as  $A++$

binary operators adds two things  
such as  $A+B$

Programming happens from top to bottom



Variable is like a storage box that store the value ~~not~~



$A = 5;$   
 $A = 6;$   
 $A = A + 6;$

A variable has no need of either single quote  
or double quotes

< script >

document.write ("A red");

window.alert ("A red")

write under body

to draw  
alert window  
on above body

</script>



## Type of operators

- (i) Arithmetic operators
- (ii) Assignment operator

Assignment ~~operator~~ operators such as  $=$ ,  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$ .

arithmetic operator such as  $+$ ,  $-$ ,  $*$ ,  $/$ ,  $\%$

Example 1 for Assignment operators.

int a (draw a box a)  
a = 12; — store value 12 in a box  
document.write(a); ~~print a~~ to show on browser  
output is 12, under body

### Example 2

inta (class a box a)  
a = 12; store value in a box  
a = 3;  
a = 34;  
document.write(a); ~~print(a)~~  
output is 34

~~Example 2~~

```
int a;  
a = 12;  
a = a + 4; or a += 4;  
document.write(a);
```



Example 4

```
int a;  
a=12;  
a=3;  
a=a+6 or a+=6  
printf("%d", a); document.write(a);  
output is 9
```

Relational operator or comparison operators that tell relations between two operands

if relation is true that's relation return true

if relation is false that's relation return false

such as  $<$ ,  $>$ ,  $<=$ ,  $>=$ ,  $==$ ,  $!=$

$a < b$ ; true or false

$a > b$ ; true or false

$a <= b$ ; true or false

$a >= b$ ; true or false

$a == b$ ; true or false

$a != b$ ; true or false

~~Best~~ Logical operator

Relationship between two ~~boolean~~ expression

one ~~boolean~~ expression has either true or false

such as  $\&\&$ ,  $\|\|$ ,  $!$

$a < b$

$\&\&$

$a == b$

boolean operator or logical operator

this operator decides

on decision this also return true or false

this also may either true or false

operator by a false boolean

that's may has either true or false logical and



boolean table  
expression 1

T	T	T
T	f	f
f	T	f
f	f	f

expression 2  
and decision in behalf  
of both expressions

logical operator expression  
- and

expression 1

$a \leq b$

$a == b$

expression 2

T	T	T
T	F	T
F	T	T
F	F	F

expression 2  
or decision in behalf of  
both expressions

logical or

increment operator or decrement operator (unary  
operator)

suffix ++, --

Example 1 (Post-increment)

int a;

a = 12;

a++;

document.write(a);

~~Printed 12~~  
Printed 13

Example 2 (Pre-increment)

int a;

a = 12;

document.write(a); ++a;

~~Printed 12~~  
Printed 13



output is 13

Example 3 (Post Increment)

int a, b;

a = 12;

b = a++;

~~document.write(b);~~ ~~print ("val of a is " + a);~~

~~document.write(a);~~ ~~print ("val of a is " + a);~~

output is b = 12

a = 13

Example 4 (Pre increment)

int a;

a = 12;

b = ++a;

~~document.write(a);~~ ~~print ("val of a is " + a);~~

~~document.write(b);~~ ~~print ("val of a is " + a);~~

output is

a = 13

b = 13

In Javascript we use document.write(a);  
for variable, document.write("abc"); for string,  
document.write('a'); for character  
for including output in the above code for  
what we performed previous

such as

A = 5;

A = 6;

A = A + 6;

document.write(A);



$a = 5;$   
 $a++;$   $b$  (increment by 1)  
 $++a;$   $b$  (increment by 1 in  $a$ )

$a = 6;$   
 $b = a++;$   
 ↓  
 that's assigning operator  
 that's means  $b$  has value of  $a++$

$a = 7$  because in second line  $a$  is represented as  $a++$

$b = 7$

$a = 6;$   
 $b = a;$   
 $a = 6$   
 $b = 6$

$\%$  is modulus that's used in place of remainder such as

$$2 \overline{) 5} \begin{matrix} 2 \\ 4 \\ \hline 1 \end{matrix}$$

remainder is 1, modulus is 1

$\Rightarrow$  that's means we equaling the things such as  $a == b$   
 if  $a = 2, b = 2$ , that's a both to be



equal

This also called equality

Not operator (!)

used for converting false to true or true to false

such as

$a = 6;$

$b = 7;$

$(!(a > b))$

that's condition to be false  
but you applied not on that

so finally you got true

OR operator for or operator one of the condition must be true then whole to be true

such as

$a = 7;$

$b = 6;$

$a > b$

||

$b > a$

that's true

that's false

so finally you got true

And operator for and operator both of the conditions must be true then whole to be true such as

$a = 7;$

$b = 7;$

$a > b$

||

$a < b$

true

false

so finally you got false



$a = 6;$   
 $b = 4;$

$a = a + b;$  (increment a by adding a and b.)

$a = 10$   
 $b = 4$

$a += b;$  (increment a by adding a and b)  
 $a = 10$   
 $b = 4$

Write a program to find maximum between two numbers

$a = 12;$   
 $b = 13;$  (these values may be anything you don't need to worry about) so forget about that after writing

$\text{if}(a > b) \{ \text{document.write}("a \text{ is greater"} + a);$   
 $\text{if}(b > a) \{ \text{document.write}(b + "b \text{ is greater}");$

There are two numbers so 2 possibilities, one be  $a > b$ ,  $b > a$  so we do write both

There is a strategic way that after condition written, you have to write  $\text{document.write}$  string according to the condition

3  
2  
1  
2  
2



such as here saying a greater than b  
`if (a > b) { document.write ("a is greater"); }`

write a program to find maximum between three numbers

if a box has 3 things, that if one is greater than other must be greater than other of the two things such as  $a(a > b \text{ and } a > c)$

`a = 12;  
b = 13;  
c = 14;  
if (a > b && a > c) { document.write ("a is greater"); }`

`if (b > c && b > a) { document.write ("b is greater"); }`

`if (c > a && c > b) { document.write ("c is greater"); }`

You can take numbers of if's at the same time note that if you write else after third if means that it belonged to that third if means if else third if to be true then that's else to be false

write a program to check whether a number is negative, positive or zero



that a here belong to negative numbers

```

a=12;
if(a<0){ document.write ("a is negative");
}
// that a here belong to positive numbers (above 0)
if(a>0){ document.write ("a is positive");
}

```

Write a program to check whether a number is divisible by 5 and 11 or not

```

a=11;
if(a%5==0){ document.write ("a is divisible by 5");
}

```

%) % means we have first here divide a by 5, then we get remainder a modulus recognised by % sign  
for perfect division there must be no remainder  
if  $a=5$ ;

$$\begin{array}{r} 5 \overline{) 5} \\ \underline{5} \\ 0 \end{array}$$

here remainder is zero  
so, here a is divisible by 5

for clearly we use  $a \% 5 == 0$  for surety that that's remainder must be equal to 0

Write a program to check whether a number is even or odd

```

a=9;
if(a%2==0){ document.write ("a is even");
}

```



if a number divisible by 2 that's, be even

if a number divisible by 3 that's, be odd

write a program to check whether a year is leap year or not

$a = 2024;$   
if  $(a \% 4 == 0 \text{ \& \& } a \% 100 != 0) \text{ or } (a \% 400 == 0)$  <sup>have not divisible by 100 so we have here!</sup>  
document.write("a is leap year");  
}

~~The year which is divisible by 4 that's a leap year and a year that's not divisible by 100~~

in the case of  $!$  sign we don't need to use  $==$  double equals as equality we can use single equals too such as  $!=$

Writing a string and variable in a same document

write("string" + ( <sup>here is variable</sup> ));

document.write("string" + a);

output is

for space add space in string such as

document.write("string" + a);

output is

string a

here is space



In some cases you have to calculate the things first then apply conditional if on it such as

you want to show calculate as a output for that ~~you~~ do before calculations

$note1 = a / 2000;$

document.write("note1" + note1);

That's a variable we create for assigning calculations

If you want to update anything value then say this variable name at left side then add this on right side too

$a = a / 2000;$

and do whatever you want

So now  $a = 1000$  after completion of this  $a / 2000$  calculation

So now if you ~~can~~ create ~~it~~ if then that's belong to updated  $a$

$\%$  modulus uses for gaining remainder or use that too

$|$  division uses for gaining quotient or use that too

In some cases we use three step conditional



~~if ( ) {~~  
~~then else if~~  
~~if ( ) {~~  
~~else~~  
if ( ) { — step 1  
}  
else if ( ) {  
} — step 2  
else { — step 3  
}

if ( ) to be false then give else if follows ~~then else if~~

If again else if goes false then else to be followed

Write a program to check whether a character is alphabet or not

```
a = 'A';  
if (a >= 'A' || a <= 'Z' || a >= 'a' || a <= 'z') {  
    document.write("character is alphabet");  
}  
else {  
    document.write("character is not alphabet");  
}  
}
```

→ This not sufficient to alternative this on next page

Write a program to input angles of a triangle and check whether triangle is valid or not

```
a = 60;  
b = 60;  
c = 60;  
if (a == b && b == c && c == a) {  
    document.write("triangle is valid" + a + b + c);  
}
```



Write a program to check whether character is alphabet or not

```
X = prompt("Enter any input");
if (X >= 'a' && X <= 'z' || X >= 'A' && X <= 'Z')
    document.write("character is alphabet");
}
else {
    document.write("character is not alphabet");
}
```

Write a program to input angles of a triangle and check whether triangle is valid or not

```
a = 60;
b = 60;
c = 60;
if (a == b && b == c && c == a) {
    document.write("triangle is valid" + a + b + c);
}
else {
    document.write("triangle is not valid" + a + b + c);
}
```

Write a program to input angles of a triangle check whether triangle is valid or not

```
a = 40;
b = 60;
c = 80;
if (a + b + c == 180) {
    document.write("triangle is valid");
}
else {
    document.write("triangle is not valid");
}
```



write a program to check whether the triangle is equilateral, isosceles or scalene triangle

```
a = 60;
b = 50;
c = 40;
if (a + b + c == 180) {
    document.write("equilateral triangle");
}
else if (a == b || b == c || c == a) {
    document.write("isosceles triangle");
}
else {
    document.write("scalene triangle");
}
```

Write a program to calculate profit or loss

sp = 500;

sale price

pp = 600;

purchase price

```
if (pp > sp) {
```

loss = pp - sp;

document.write("loss" + loss);

}

```
if (sp > pp) {
```

profit = sp - pp;

document.write("profit" + profit);

}

```
if (sp == pp) {
```

document.write("no profit or loss");

}



Write a program to find month name by  
enter first letter of month  
X = prompt("Enter month first letter");  
if (X == 'J' || X == 'j') {  
document.write("January");

}  
if (X == 'F' || X == 'f') {  
document.write("February");

}  
if (X == 'M' || X == 'm') {  
document.write("March");

}  
if (X == 'A' || X == 'a') {  
document.write("April");

}  
if (X == 'M' || X == 'm') {  
document.write("May");

}  
if (X == 'J' || X == 'j') {  
document.write("June");

}  
if (X == 'A' || X == 'a') {  
document.write("August");

}  
if (X == 'S' || X == 's') {  
document.write("September");

}  
if (X == 'O' || X == 'o') {  
document.write("October");

}  
if (X == 'N' || X == 'n') {  
document.write("November");



```
if (x == 'd' || x == 'D') {  
    document.write("December");  
}
```

Write a program to input any alphabet  
check whether it is vowel or consonant

```
a = 'a';  
if (a == 'a' || a == 'e' || a == 'i' || a == 'o'  
    || a == 'u') {  
    document.write("vowel");  
}  
else {  
    document.write("consonant");  
}
```

Write a program to check whether a character  
is upper or lowercase

```
a = 'd';  
if (a >= 'a' && a <= 'z') {  
    document.write("lowercase");  
}  
if (a >= 'A' && a <= 'Z') {  
    document.write("uppercase");  
}
```

Write a program to check whether character is  
alphabet, digit or special character

```
a = '@';  
if (a >= 'A' || a <= 'a') {  
    document.write("alphabet");  
}
```



```
if (a <= 0 || a >= 10) {
    document.write("digit");
}
```

```
if (a == '#' || a == '@' || a == '$') {
    document.write("special character");
}
```

Write a program to find input marks of 5 subjects

```
X = prompt("Enter marks");
```

```
if (X >= 450 && X <= 500) {
    b = X / 500 * 100;
```

```
document.write("Grade A" + b);
}
```

```
if (X >= 400 && X < 450) {
```

```
    b = X / 500 * 100;
```

```
document.write("Grade B" + b);
}
```

```
if (X >= 350 && X < 400) {
```

```
    b = X / 500 * 100;
```

```
document.write("Grade C" + b);
}
```

```
if (X >= 300 && X < 350) {
```

```
    b = X / 500 * 100;
```

```
document.write("Grade D" + b);
}
```

```
if (X >= 200 && X < 300) {
```

```
    b = X / 500 * 100;
```

```
document.write("Grade E" + b);
}
```

if (X >= 450 && X <= 500) {  
 b = X / 500 \* 100;  
 document.write("Grade A" + b);  
}  
if (X >= 400 && X < 450) {  
 b = X / 500 \* 100;  
 document.write("Grade B" + b);  
}  
if (X >= 350 && X < 400) {  
 b = X / 500 \* 100;  
 document.write("Grade C" + b);  
}  
if (X >= 300 && X < 350) {  
 b = X / 500 \* 100;  
 document.write("Grade D" + b);  
}  
if (X >= 200 && X < 300) {  
 b = X / 500 \* 100;  
 document.write("Grade E" + b);  
}



Write a program to find notes in given amount.

```
X = prompt("Enter amount");
if (X >= 2000) {
    not2000 = X / 2000;
    document.write("not 2000 = " + not2000);
    X = X % 2000;
}
if (X >= 1000) {
    not1000 = X / 1000;
    document.write("not 1000 = " + not1000);
}
```

Switch Syntax

ch =  $\rightarrow$  anything placed here either number or character

switch (ch)  $\rightarrow$  that's user defined

that's ~~predefined~~  
predefined

That's maybe x, y, z, a, b,  
 $\hookleftarrow$  such that

ch = 's'; character

switch (ch)

we assigning user defined value

// next line

comment

how to create cases

case 1:

```
write here to what to show as output
document.write("one");
break;
```

break is necessary for  
so that it cannot start next case here  
if it is case 2, then 3 and further default



Case 2:

```
<
document.write("Two");
break;
}
```

Case 3:

```
<
document.write("Three");
```

Case's:

```
<
document.write("Sunday");
break;
}
```

default:

```
<
document.write("Invalid choice choice");
break;
}
```

if you can't create any case then it automatic  
that default case is executed.

Prompt

```
X = prompt("Enter anything");
```

actually in to write prompt we use  
thing that's why



after entering anything we get string too  
maybe you get 3 that's be string but  
a number  
enter variable here to convert  
to number  
so sometimes we use `parseInt()`

such as

`y = prompt("Enter anything");`  
that's variable  
here we enter anything  
may be number  
that's store in a  
y variable

Now convert that it into number, that's a  
string because

`y = parseInt(y);` here write variable  
to which convert  
that's variable here to update y value

Now i do check whether it is number or  
string?

so i write

`document.write(" " + typeof(X));`

we use `<script></script>`

object for using

javascript in html

for using in html we use string quotation is  
'javascript'

that's shows in  
output that x is  
number or string



if i want to use ~~break~~ in Javascript  
of html

document.write("<br>" + space(X));

function

like if you want to do number of  
tasks then you write code according to  
that then you want to again do that  
tasks so what you do its better to use  
that code which you first written <sup>rather than</sup> for  
that use function

function is actually a set of  
code

working  
again and  
again

Pillars of functions are (1) Body

(2) function calling

that's take time

and your project  
be bigger too

function body written can't work without function  
calling

function body

key word

// new line

function functionName ( )

code be written here such as  
document.write("one");



function calling

functionName();

such as

```
function a() {  
    document.write("One");  
}  
a();
```

function Parameters these are ~~actual parameters~~  
of two types Actual parameter and formal  
parameters

formal parameter applied to <sup>function</sup> Body  
Actual parameter applied to function calling  
Both of them applied under ( ) square braces  
such as

function functionName( ) { } formal Parameter

value write here

functionName( ) Actual Parameter

formal Parameter is actually a variable may be  
x, y, z, a, b, c

Actual Parameter is actually a formal parameter  
variable value such as 1, 2, 3, One



such as

```
function functionName(x, y) {  
  document.write(x+y);  
}
```

functionName(1, 2);

output be 3

```
function functionName(x, y)  
{  
  document.write(x);  
  document.write(y);  
}
```

functionName(1, 2);  
output be 12

```
function functionName(x, y)  
{  
  document.write(x);  
  document.write(y);  
}
```

functionName(1, 2);  
Output be 11



Write a program to find input marks of 5 subjects

```
X=prompt("Enter Marks");
b=x/500*100;
if (b>=90 && b<=100)X
document.write("Grade A"+b);
}
if (b>=80 && b<90)X
document.write("Grade B"+b);
}
if (b>=70 && b<80)X
document.write("Grade C"+b);
}
if (b>=60 && b<70)X
document.write("Grade D"+b);
}
if (b>=40 && b<60)X
document.write("Grade E"+b);
}
```

function return it's just used after successful done of tasks after calling it's just for using to do a return or message that that's tasks are completely done

That's used in body after the code which you already written function

That's actually replaces function name by return value

Such



```
function functionName(x,y)
{
    document.write("One");
}
```

functionName(1,2);

So Now a task is done

```
function functionName(x,y)
{
    document.write("One");
    return "yes";
}
```

Not a function returns

```
x = functionName(1,2);
document.write(x);
```

A function has one return such as

```
function a(x,y)
{
    if(x%2==0){
        return "Even";
    }
```

```
    if(y%2==0){
        return "Even";
    }
```

```
    if(x%2==0){
        return "Odd";
    }
```



if  $(y \% 3 == 0)$  ✓  
return "odd";

↗ here x is odd, y is even

x = a(3, 2); for that  
document.write(x); ✓

conditions with  $x \% 2 == 0$   
,  $y \% 3 == 0$  to be  
true

~~if there is output is first odd or~~  
~~even in y as return then odd is~~  
~~a function has one return~~

~~if you get one return then other return~~  
~~is not possible~~

So there are two returns and shown as output too  
it's automatic that ~~programming~~ first return to be chosen first as programming  
goes from top to bottom and the other return  
to be remain same

But if you ~~commented~~ commented one already  
chosen return then ~~the~~ return goes a work  
and shown as output ~~other~~ return first

You can check the return by document.write  
(x);



## DOM (Document Object Model)

That means  
get the element or either object  
if you get that object in console you get  
not object  
if not you get that object in console you  
get null

What is document as here such as

`<h1> This is Rohit </h1>`

get's whole a document

The written thing between the `<h1>` `</h1>`  
tags this is innerHTML.

if you want to update innerHTML which you  
already wrote then  
write first here variable (here x)

This has document.`getElementById("h1 tag");`  
that means x has h1 tag string stored, now you  
want to update that

so `x.innerHTML = " "`

want to update

write here what you

DOM works is to get the document  
element (such as h1 tag and more), by specified  
Id, id is actually written on the object or  
element to whom which to get



such as  
in HTML

document

```
<body>  
  <div id="div">  
  </div>  
</body>
```

id="div"

this id created here

want to get the div element but first  
is document.

Now Javascript part

```
<script>  
x = document.getElementById("div");  
you can use you can use console.log(x);  
or may use document.write(x);  
use
```

↓  
for check what  
has x stored

```
</script>
```

Now you want to update innerHTML

```
x.innerHTML = "OK";
```

string write here do which  
update  
here it is OK

onclick event is used to call the function as  
per we need it generally linked to button  
tag

```
<button onclick="function calling here">button name here</button>
```



~~script~~

```
function abc()
{
  x=document.getElementById("div1");
  console.log(x);
  x.innerHTML="OK";
}
```

function body

~~function calling~~  
here is conditional

<button onclick="abc()" >button text here / </button>

here onclick function to be call

<body> <div id="div1"> here it is what </div> </body>

~~basically focus on first~~

window

~~is~~ is global object

for gaining other global objects in console only  
we use keyword window

x=window;

on console, tap > console.log(x);  
global object window for gaining list of  
version



Selection statements are of three types

- ① if
- ② if-else
- ③ else-if

Loop is continuous repetition of <sup>some</sup> code, that's automatic by conditional operator such as greater than, less than or equal to

Loop of three types

- ① for loop
- ② while loop
- ③ do while loop

Step 3 goes on

Three of both works on same steps such as

For loop

its code written in one line

Syntax for (initialization, condition, increment/decrement)  
statement to be written here

such as

```
for (i = 1; i <= 10; i++)
```

```
    document.write("One");
```

```
}
```

- ① initialization
- ② condition (if true then)
- ③ Statement execution which to be written under condition
- ④ Increment / decrement
- ⑤ Again check condition
- ⑥ continuous goes on

while loop its code written in column such as

Syntax ~~while~~ initialization, while (condition here)



✓ Statement to be written here  
Increment

✓ Statement to be written here:  
// new like Increment / Decrement

fox and white <sup>deep</sup> almost ~~blue~~

## do while loop

le l'année

individualization

keyword

~~do~~ ✓

✓ statement to be written here

Statement to be submitted  
/ ~~File~~ Line Increment / Decrement

keyword while (condition here);

break and continue

Actually ~~break~~ break keyword is used to stop the loop at specific condition.

Continue is to skip that specific condition to whom continue keyword is applied

Confirm box and Alert Box

confirm box is used to <sup>do</sup> confirm anything

That's you  
confirming

This page says

~~if you signed up~~

○K

Case

That's 600 with  
keyword

~~therefor false the~~  
condition

Therefore true condition



sting written here

Green House

Date

Page No.

$x = \text{confirm}(" \wedge ");$  here it is

Is you signed up

confirm box based on true or false so conditional if else must be used

if (x) Is you signed up

when user click ok then x value matches the user defined value so if to be true

if (x)

document.write("Is you signed");

}

else

{

document.write("You're not signed up yet");

}

for loop

for (i=1; i<=10; i++){

document.write("One" + i + "<br>");

}

output is

One 1

up to

One 10

Now ~~you~~ you want to break the loop at  $i=5$  then

INDIA

2019

Classmate India's No. 1  
notebook brand is now a Superbrand  
Survey conducted by NIELSEN in Jan 2019.



Now write the if ~~as~~ statement  
then use break keyword in next line

```
for(i=1; i<=10; i++)  
{  
    document.write("One");  
}  
break;
```

~~Output~~  
~~One~~  
~~One~~  
~~One~~  
~~One~~  
~~One~~

If you want to skip the loop at some specific condition

```
i=4  
for(i=1; i<=10; i++) {  
    document.write("One");  
    if(i==4)  
        continue;  
    document.write("One" + "<br>");  
}
```

one  
one  
one  
one  
up to  
one

In break and continue

document.write("Skip here");  
Keyword continue and break



with break keyword

Green House  
Date \_\_\_\_\_  
Page No. \_\_\_\_\_

```
for (i=1; i<=10; i++) {  
    if (i==5) {  
        break;
```

```
    }  
    document.write("One" + i + "<br>");
```

→ That means we taking the output after using the keyword break

output is

One1

One2

One3

One4

while loop

```
i=1;  
while (i<=10) {
```

```
    document.write("One" + i + "<br>");  
    i++;  
}
```

output is

One1

One2

One10



while loop with break keyword

```
i = 1;
while (i <= 10)
{
    if (i == 5) break;
    document.write("One" + i + "<br>");
    i++;
}
```

Output is

```
One1
One2
One3
One4
```

do while loop

```
i = 1;
do {
    document.write("One" + i + "<br>");
    i++;
} while (i <= 10);
```

Output is

```
One1
One2
One3
One4
```

do while loop with break keyword



```

i = 1;
do {
    if (i == 5)
        break;
    document.write("One" + i + "<br>");
    i++;
} while (i <= 6);
    
```

### Events

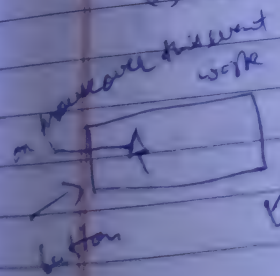
- (1) Click Events
- (2) Key Events
- (3) Mouse Events
- (4) (i) onfocus event  
(ii) onblur event

#### (1) Click Events

- (1) onclick → on single click this event work
- (2) ondblclick → on doubleclick this event work
- (3) oncontextmenu → on rightclick this event work

#### (2) Mouse Events

- (1) onmouseover
- (2) onmouseout
- (3) onmousedown → on single click this event work
- (4) onmouseup
- on mouseout this event work but first we do mouseover then his works



#### (3) Key Events

- (1) onkeydown
- (2) onkeyup
- (3) onkeypress

on mouseup this event work but first we do mousedown but when we left mouse down from click position this event to be work





these events are actually works with input tag

### (3) Key Events

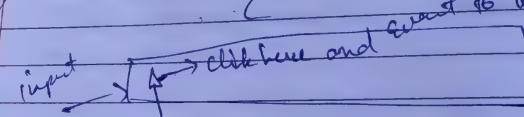
- (1) onkeydown → on key press this works  
→ keys of all types (Alphabet, number, special character, functional keys such as ctrl, Alt and more)
- (2) onkeyup
- (3) onkeypress → on key press this event works with (Alphabet, number, Special character but not with functional keys)

This event works with first we do keydown but do keyup free when we left click key from click position this event do be work

### (4) (1) onfocus Event

→ This works with when you do focus on something in input by clicking inside input tag

These events works with input



This event work with when you do first on focus Event and then click outside the input that's a blur Event

### (5) Load Events

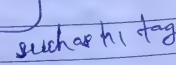
These events are work with body

- (1) onload Event
- (2) onresize Event
- (3) onunload Event

on successful page load this event works

→ on resizing the page this event works by when we do resizing the browser size





~~Remember one thing~~  
X-style-background = "green"; {script}  
↓  
not for background color input tag id  
That must body tag  
it manually

Update the onclick event ~~to point~~ <sup>to</sup> ~~the~~ <sup>the</sup> href attribute  
after a successful onclick or single click  
what's means we updating the href through  
onclick event itself for this onclick  
this keyword

Classmate India's No. 1\*  
notebook brand is now a Superbrand.  
\*Survey conducted by ASSOCHAM in Jan 2019.



button onclick="this.innerHTML='Signup'"

single quotes here  
because double quotes  
are already used as  
string.

under the curly braces whatever is written that  
is local to that you can't get access outside  
these boundaries, so for that any to element which  
you have to first get that element into  
your local or under curly braces then  
you can update that

id keyword here

```

<
x = document.getElementById("id");
x.innerHTML = "One";
}

```

### Datatypes

- (1) Number [all types of numbers such as  
real numbers, irrational numbers and more]
- (2) String → ' ' single quotes, " " double quotes,  
" " " triple quotes  
if you writing another string under one string  
then use single quotes for that  
" ( ) "
- (3) Boolean → True, false
- (4) Null → Null can't accept 0  
x = null then you can't update x too
- (5) undefined → x = undefined that's mean



That you not get defined  $x$   
may be later you can do define that  
such as by update the  $x$  by some  
value

(6) Array that stores multiple elements  
such as  $X = \boxed{10}$  — element 2  
 $\quad\quad\quad \swarrow$   
 $\quad\quad\quad$  here is two elements  
 $\quad\quad\quad \searrow$   
 $\quad\quad\quad$  element 1

elements are stored by index, index starts from 0  
syntax  
variable here =  $[1, 2, 3, 4, 5, 6]$ ;  $\swarrow$  base  
 $\searrow$  braces  
index  $\swarrow$  then are six elements

Number of elements defined array size to here  
array size is 6

index find by this formula,  $O = n - 1$   
 $\downarrow$   
 memory

index find by this formula  $i = \frac{n}{2}$   $\downarrow$  number of elements

$\therefore$  here element stored at 0 index

if we do  $\times [q]$  -

if we do  $X[9]$  index  
then we get 1, that is element which  
is stored at index 0  
That means ~~array~~ we checking here element  
at index 0 and ptr's do  
we stored in variable x here

→ That means  $x$  here  
be stored in variable  $x$  here



(2) Object → That's cover key/property and their value  
syntax →

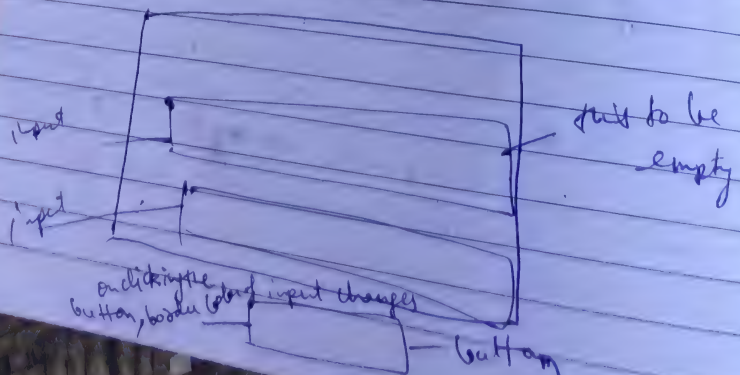
Variable here = ✓  
Key/property here : value  
name : "Rohit",  
class : "abc",  
Roll No : "xy2",

}  
you can write here number too

if you want to check name property  
then

variable here . name  
that's give you name value at  
console

Validation works on input tag when you not typed  
anything input border to be turn red that's  
error validation that's means if user not typed  
get and highly to ~~login~~ login  
username and password inside input tags





```

id = "input1"
<body>
  <button onclick = "abc()" > click me
  <input type = "text" placeholder = " " / button>
  <script>
    function abc()

```

```

      x = document.getElementById("input1");
      y = document.getElementById("input2");
      if (x && y == " ") {

```

document.  
getElementById("input1")  
paragraph

```

      id = "input2"
      <input type = "password" placeholder = "password" />
      <p id = "paragraph"> </p>

```

```

      x.style.border = "2px red solid;";
      y.style.border = "2px red solid;";
      z.style.color = "red!";

```

```

      z.innerHTML = "You not write anything yet!";

```

```

    } else {
      x.style.border = "2px green solid;";
      y.style.border = "2px green solid;";
    }

```

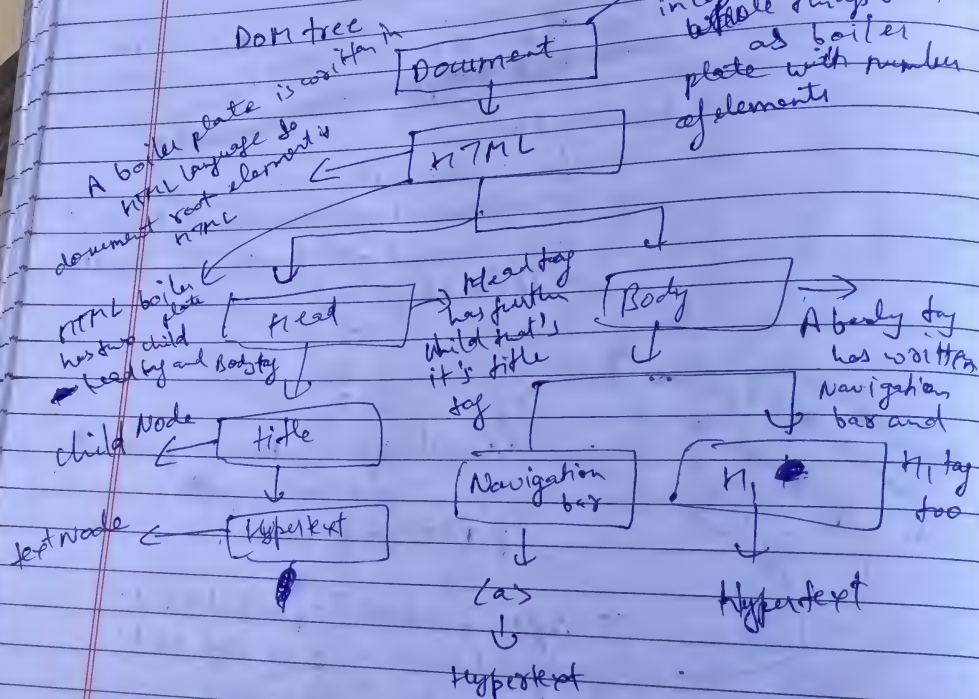
In the final we set the paragraph by input 1 to be empty or not write yet may be update by innerHTML in javascript



India - India's No. 1  
now a Superbrand  
in 2019



This is complete file in which we write ~~whole~~ things such as boiler plate with number of elements



we ~~are~~ targeting here document  
on targeting that we get complete boiler  
plate which you write with elements or tags

body < /body>

or a complete document

< /DOCTYPE html>

< body>

< /body>

< /html>



## Target Types

- (1) Target By Id
- (2) Target By class
- (3) Target By Tag Name

### (1) Target By Id

Id is unique ~~id~~, so that's unique for every element. Targeting means we getting the target thing here that's element

Targetting or getting the element by Id  
document.getElementById("Id be written here");

### (2) Target By class

Here it's not unique id, here is class ~~id~~  
one class may linked to number of ~~class~~ elements  
so we use here term called elements

Targetting or getting the elements by class  
document.getElementsByClassName("class name here");  
↓ that's standard

### (3) ~~Targeting~~ Target By Tag Name

Here is the case of Tag Name  
one tag ~~id~~ may used number of time  
so we use here called elements too

Targetting or getting the elements by Tag Name  
document.getElementsByTagName("Tag Name written here");



Classmate - India's No. 1<sup>st</sup>  
brand is now a Superbrand.  
by ASSOCHAM in Jan 2019.



```
console.log(document.all);
```

document: all);  
That's give you boiler plate or document  
tags in a order way in you created

This (document.all) give your document  
in a ~~tags~~ <sup>(key names)</sup> ~~order way~~  
~~you either tags such as head body has child style city~~

So basically it's give a overview <sup>of</sup> <sup>by key names</sup> tags ~~not complete~~  
such as html, head, body

document.all give ~~that key~~ in a array form

↑ This keyword for taking document output with document. all using square brackets

HTML All Collection (8)

(html, head, meta, title, script, element, document, tags, key names, boilerplate)

index 0: html  
index 1: head  
index 2: meta

This saying how much number of elements used

boiler plate

~~Don't keep track~~  
~~Days from hanging~~  
~~have been much~~  
~~distraction of day~~  
~~has been that~~  
~~accounting and~~  
~~friends which~~  
~~attained is not~~  
~~possible~~

~~<html>~~  
~~<head>~~  
~~<meta>~~  
~~<title>~~  
~~<script>~~



Variable stored in a array at the different indexes

so we get. ~~big~~ index of the elements  
too by writing the console.log(document.  
all)

```
console.log(document.all[5]);
```

we getting here pay bag which is stored at index 5

console = log (document.links);

key words may be  
link, links, best URL  
or more

we gettings here links if anyone added  
if not you created them for family downe  
you post output with document.

for taking document  
output with  
document. (✓)

document-links

mostly maybe

mostly ~~link~~ if you  
have a link or more length: 0  
high PRI has higher

section [ ] works only with keys  
cant put is if you added multiple links  
narrow collection (2)

baseURI has type (1) link

1.  $\text{if } \text{is\_url\_collection}(i)$   
     $\text{index} \leftarrow \text{URLCollection}(i)$   
     $\text{if } \text{is\_anchor}(i)$   
        $\text{if } \text{is\_anchor}(i)$   
        $\text{length} \leftarrow 2$

parse by (document.baseURI); gives you <sup>length: 2</sup> right now webpage html path such as  
http://127.0.0.1:5501/login.html

```
console.log(document.links);
```

mode-log (document-link),  
that's give you undefined because you  
not created link here details about webpage  
which you use

create by (document.location); give you a link ~~that~~ right now  
origin, pathname that's about  
such as "/login.html"

st. location); give you that's about  
origin, pathname such as "/login.html"



```

</body>
</script>
x = document.getElementsByClassName("div");
console.log(x);
<div class="div">
  </div>
<div class="div">
  </div>
</body>

```

Output is HTMLCollection []

index 0 → div div  
 ↓  
 div element      that element has linked class name that is div

If you stored another get element in same variable in which you previously done such as here is x.

work done work  
 (X = document.getElementsByClassName("div");  
 now stored another element in that variable)

X = document.getElementsByTagName("div");  
 output is

HTMLCollection []

0: div div      that's div tag  
 1: div div      that's class linked  
 ↓      that's class linked too  
 length: 2 (because here two elements or tags)



## Reactjs

- ① Library of Javascript
- ② open source (you don't need to buy subscriptions to use them or paid them)
- ③ used for create UI (user Interface) of web application  
It's just for viewing to customers not properly work at all.
- In some way it's a frontend just
- (4) Approach it's to just view or for visual the website and
- (5) frontend language other functions do be not working at all

### Library Reactjs

has inbuilt functions

It has view (or V)  
It just direct way to view

### framework

Angular's framework of Javascript

has inbuilt functions

It has MVC (Module view controller) such as process starts from module to then controller to then view

### NPM

It's be used in particular project which you working on

### NPM

It's be used Globally by CMD (command)

It's a Node Package Executor or Runner

It's a Node Packages manager



Classmate - India's No. 1  
notebook brand is now a Superbrand  
as declared by ASSOCHAM in Jan 2019.



NPM for  
executing the  
downloaded  
packages (by NPM)

NPM for download or  
install the packages

for proper functioning of the Reactjs  
you must install Nodejs in your  
device

after successfully installed check that install or  
not by  
open command and type node -v

for node version

for successful run of the Reactjs projects  
must install NPM globally

npm install -g create-react-app  
global

after successfully installed check that install  
or not by  
open command and type npm -v

for npm version

Newest way to open project folder  
Right click that folder and click open in  
terminal and type code - . & dots and then  
open, you got visual studio code to be  
open



Now click terminal and type `npm create-react-app`  
"project Name here"

This will create a project to the specified  
path or folder React.js

`cd` is the shortcut key for move  
such

`cd desktop` (used for move to  
desktop)

`npm start` is the abbreviation for start or  
run the project which you right created or  
view that at the local host

React.js is a single page application because  
there is ~~not~~ need of go live or need of  
refresh the page but still if you modifying  
the sources changes so be seen at the view  
or at the web application

If you downloaded packages or install one that's  
done under the node-modules folder in your  
project

so do of the work we done in the App.js  
in the react.js projects

Project dependency based on node-modules folder  
that's contain resources used for your react project  
for successful run



Classmate India's No. 1  
Book brand is now a Superbrand  
Awarded by ASSOCHAM in Jan 2019



we first <sup>install</sup> "npm install -g" by keyword here "

↓  
create-react-app

we may use this keyword to access  
install or downloaded packages  
during project

Then we ~~use~~ use npm packages to particular  
project

npm "npm keyword here" "here project name"

keyword  
for accessing or  
running the  
npm packages  
at your  
own project

↓  
create-react-app

npm install "Package name" <sup>here</sup>  
install that package in your node-modules



## DOM in advance

### DOM GET

- (1) innerText
- (2) innerHTML
- (3) GetAttribute
- (4) Attributes
- (5) GetAttribute Node

### DOM SET

- (1) innerText
- (2) innerHTML
- (3) Set Attribute

### DOM GET and SET

- (1) innerText

<div>

ch<sub>1</sub> id="h<sub>1</sub>">hello</h<sub>1</sub>>

</div>

text between tags

↑  
<h<sub>1</sub>>innerText  
</h<sub>1</sub>>

This shows innerText of the element (target)  
↑ By id specific

<script>

x = document.getElementById("h<sub>1</sub>").innerText = "hello";

updated text written here

↓  
whole its a innerText SET

console.log(x);  
</script>

### (2) innerHTML

<div>

ch<sub>1</sub> id="h<sub>1</sub>">hello</h<sub>1</sub>>

</div>



Classmate India's No. 1  
ask to find is now a Superbrand  
led by award in Jan 2019







In console you get value of that attribute

```

set Attribute
<div>
  <div class="class" id="id" name="name" style="border: 2px solid red;">
</div>

```

```

<script>
x=document.getElementById("Id be written here");
after set attribute we must to get attribute again
x=document.getElementById("attribute name written here");
console.log(x);

```

### Get Attribute Node

Get Attribute gives you a attribute value  
But by Get Attribute Node, you get attribute with value

```

<div>
  <div class="class" id="id" name="name" style="border: 2px solid red;">
</div>

```

```

<script>
x=document.getElementById("Id be written here");
getAttributeNode("attribute name be written here");

```

</script>



Classmate - India's No. 1\*  
Notebook brand is now a Superbrand.  
\*As per survey conducted by NIELSEN in Jan 2019.



output is  
name = "name1"

To check how many numbers of attributes  
linked to your ~~for~~ element

<div>  
<h1 id="h1" name="name1">hello </h1>  
</div>  
this may be to get numbers of  
attributes

<script>  
x=document.getElementById("h1").attributes;  
console.log(x);  
</script>

its a keyword to get  
numbers of attributes  
to your element

output you get in array form

NamedNodeMap(0: id, 1: name, id: id, name: name,  
0: id  
1: name  
id: id  
name: name  
length: 2)

after get attributes we can get value to  
the specific attribute by ~~the~~ index of array  
name attribute at index 1

attribute with



```
<script>
```

```
x=document.getElementById("h1").attributes
```

[ ]  
array

you get

name: "name"

index of element  
to whom you  
can get be  
written here  
attribute and its

Remove attribute, that remove the value of too  
~~the attribute~~ to whom this  
applied

```
<div>
```

```
h1 id="h1" name="name">hello</h1>
```

```
</div>
```

```
<script>
```

```
x=document.getElementById("h1").removeAttribute  
("attribute name  
written here");
```

```
console.log(x);
```

so you get undefined as output on console

CSS selector are of three types you already  
studied in CSS mostly of course in internal CSS

(1) ~~id~~ selector → put # hash sign under



style tag, that's recall about Id selector

Syntax

`<style>` Id selector notation  
# " " " "  
/style V then id be written here  
name

(i) class selector → put . dot under style tag, that's recall about class selector

Syntax

`<style>`  
class selector notation  
" " " " "  
/style V then class name be written here

(ii) Tag selector → put tag keyword under style tag, that's recall about tag selector

Syntax

`<style>` tag keyword or name  
div " " " " "  
/style V then tag name written here

its alternative

querySelector to document.getElementById(" " " ")  
Id be written here

~~querySelector~~ only to get whole element  
to get the element.



querySelector with the usage of tag selector

```
x = document.querySelector("h1")
console.log(x);
```

tag selector be written here with prefix # sign such as

#h1

This give you whole element as output by which you applied Id by some value

Id value linked to element by whom we accessing here

che id="h1" name="name1" style="border: 2px solid red" > hello </h1>  
In output you get that Id with value too

querySelector with the usage of class selector

```
x = document.querySelector(".h1")
console.log(x);
```

class selector be written here with prefix . such as dot

.h1 or class value

class name, this applied to element by which

output is

che id="h1" name="name1" class="h1" style="border: 2px solid red" > hello </h1>

output same as tag selector selector querySelector

can



querySelector with the usage of tag selector

$x = \text{document.} \cdot \text{querySelector} ("h1")$ ;  
tag name or keyword  
be written here  
sent as  $h1$

console.log (x);

and get it  
 $<h1 \text{ id}="h1" \text{ name}="hanel" \text{ class}="h1" \text{ style}="border: 2px solid red" >hello </h1>$

querySelector has automatic first element select property

such

$<h1 >hello </h1>$

$<h1 >hello1 </h1>$

there are two  $h1$  tags

but when we select by tag selector and get ~~any~~ element by querySelector you get only first element <sup>whole</sup> not second one

output is

$h1 >hello </h1>$



$\rightarrow$   $\langle h_i \rangle$  holds  $\langle h_i \rangle$

output is  $\chi(t)$

chis

only first element

element, that element has  
even that, other element

for selecting ~~the~~ all element by same keyword we use `queryselectorAll`

script 3

```

x=document.querySelector("v
if ge

```

if you say selector

your tag name or  
password be written  
here, 11);

with # hash sign

console.log(x);

scripts



first tag  
~~<h1>~~ second tag  
 <h1 id="h1" name="hanel" class="h1" style="border: 2px green solid" >hello3</h1>  
 i/h1 third tag  
 <h1>hello3</h1>

output will be the form of array  
 NodeList(3) [h1, h1#h1.h1, h1]  
 array of 3 elements  
 h1 element with id included  
 and class included too  
 there are 3 elements  
 0: h1  
 1: h1#h1.h1  
 2: h1  
 length: 3

If you want to get element at specific index  
 h1 element with index

x=document.getElementsByTagName("h1")[2];  
 console.log(x);  
 this is array index

you got index 2 element  
 <h1>hello3</h1>



~~to~~ style of types

① normal style

document.querySelector(" ").style.color  
tag selector use if  
you want be written  
such as h1 with prefix  
# hash sign for  
id selector

attribute for styling  
property  
value  
property value  
be written here

output is  
hello with red color style

here its red is the  
original in CSS is the  
background color but  
applied here

② Capitalised style

document.querySelector(" ").style.backgroundColor  
tag selector use if  
you want such as  
h1 with prefix  
# hash sign for  
id selector

property value  
be written here  
here its green

output is  
hyperlink with background color

for getting className or value by id linked to  
that element through querySelector

x = document.querySelector(" ").className; class value  
tag selector use here if want  
to want to use

console.log(x);





outputs to h,  
manipulate means update something

NPM NPMx

~~npm install~~ we borrow something from  
~~npm install~~ npm its for limited time  
for single project  
NPM downloads the create-react-app package -app such as create-react-app package and  
npm execute that too

classList keyword not says about multiple ~~class~~ ~~is~~  
classnames  
its just give you className or classvalue in  
array form

```

<h1 class="h4">
  </h1>

```

keyword to get  
className  
in array  
classList form

```

<script>
x=document.querySelector("h1")
console.log(x);

```

querySelector  
tag selector  
written here

console.log(x);  
how number of class  
name  
linked to that  
DOTTokenList(1)  
0:"h4" className  
length: 1  
value: "h4"



for getting two class at classList use  
`<h1 class="h1 h2">`  
`document.querySelector("h1").classList`  
if you want to get classList in form of array  
to about specific index

~~document.querySelector("h1")~~

querySelector be written here

console.log(x);

index number to be written here

Add className under classList

`<script>  
document.querySelector("h1").classList.add(" ");`

querySelector be written here

to whom you add class to that or applied that class properties to that

className to which you want to add

`</script>`

`<head>`

`<style>`

`h1 {`

`background-color: blue;`

`</style>`

`</head>`

hello

background-color: blue;

Remove className under classList

`<script>  
document.querySelector("h1").classList.remove(" ");`

querySelector be written here to whom you remove class to that

className be written to whom which to be remove



~~output is timer~~

callback function

→  $x = \text{setTimeout}(\text{function-name}, \text{Time in milisee})$   
- only  
1s = 1000 milliseconds or ms

$\text{clearTimeout}()$

(2)  $\text{setTimeout}(\text{function-name}, \text{Time in ms});$   
 $\text{clearInterval}()$

callback function is used to call the function on specific time such as after 3 seconds, so that's based on time property here we use  $\text{setTimeout}$  keyword for that for setting of specific time

callback function is once run only on single refresh if you want to stop the ~~the~~  $\text{setTimeout}$  then we do use other keyword called  $\text{clearTimeout}$  or  $\text{clearTimeout}$  callback function stopped at specific of time where we called such as  $\text{setTimeout}$  is goes on but sudden we use the  $\text{clearTimeout}$  at time = 3s then on  $\text{setTimeout}$  arrives time = 3s callback function stopped on

for repetition of  $\text{setTimeout}$  we may use other keyword called  $\text{setInterval}$



for stop this interval we may use clearInterval  
keyword

callback function with setTimeout  
that's actual parameter for setting time for  
keyword call the required function

(script)

x = setTimeout (function Name here, time in  
ms);

→ this variable

that's ms

but it is abc

show 3000ms

(script)

use ~~for~~ for the

3000 ms

3s

clearTimeout

or this variable

has stored

setTimeout

function abc ()

alert ("successful");

function body

So, function to be call after 3 sec

callback function with clearTimeout

what comes here

so we first setTimeout then we do  
clearTimeout

create a new function body

function abc1 ()

here to use clearTimeout (x);

alert ("cancel");

here it is x

here we will format  
parameter x as  
x = 1000  
or x = 1000  
or x = 1000

actual  
parameter  
which is x



now we to call this function here this function is  
abc1

we can do that by button tag in html

<body>  
 <button onclick="abc1()"> (Cancel / button)  
 call here  
 abc1 function

</body>

Now will stop after setTimeout  
 for repeating the callback or function to do  
 repeating the function tells till we do stop  
 even one refresh the page

callback function with setInterval  
 <script> keyword for repeating the call function  
 or calling the function  
 again and again

var x = setInterval(function-name here, Time (in ms);  
 (function abc3() {  
 alert("1");  
 })  
 here it is 3000  
 abc3

</script>

that shows  
 35 or  
 3000 ms

callback function with clearInterval

so we setInterval then we do clearInterval  
 Create a new function body

function abc4() {  
 clearInterval(x);  
 alert("stop");  
 }  
 formal parameter

variable  
 comes here



</body>

button onclick = "abc()" stop </button>

</body>

so when we click stop button this will stop the setInterval

Advance usage of setInterval

① setInterval with styling  
generally it is obvious that we do first  
use setInterval but we can have setInterval  
(script)

setInterval(functionName, time in ms);

↓  
abc

↓ 3000

function abc()

document.querySelector("v") .style.backgroundColor

class selector

= "white";

}

here  
with prefix  
such  
as  
.class

</script>

</body>

<div class = "class">  
</div>

</body>



```
<head>
<style> .class {
  border: 2px solid red;
  width: 500px;
  height: 50px;
  background-color: green;
}
```

That's mean after 2 sec  
background color goes white  
class with setInterval

Calculating and implementing in styling properties  
such as width applies according to calculation

for this accept any variable  
 $x = 0;$  // x variable assign initially to 0

Calculation, x variable increases with number 10  
 $x = x + 10;$

So, the width =  $x + 50\%$ ;  
for showing the value of x with percentage sign

setInterval (function-name here, Time in ms);  
this for one initilre  $x = 0;$  at 0 3000

function abc() {  
 x = x + 10;  
} because that's part only  
calculation here part of  
repetition

```
document.querySelector(".class").style.backgroundColor  
document.querySelector(".class").style.width = x + 50%;  
color = "white";
```



actually  $x = x + 10$ ;  
takes values first 10, then 20, 30  
~~and goes on~~  
and goes on

for showing the values of properties such as  
margin are px (pixels), width (in px)  
height (in pixels)

setInterval with the usage of styling, calculations  
and conditional if

`<script>`  
`setInterval (function-name here, Time in ms);`

initialisation

`function abc5()`

`x = 0;`  
calculation here

`x = x + 10;`

`document.querySelector(".class").style.backgroundColor = "white";`

`document.querySelector(".class").style.width = x + "%";`

`if (x == 50) {`

`document.querySelector(".class").innerText = "50%";`  
space filled up

`document.querySelector(".class").style.backgroundColor = "blue";`

`}`

`if (x == 90) {`

`document.querySelector(".class").style.backgroundColor = "red";`  
`document.querySelector(".class").innerText = "space is almost filled up";`

`}`



```

if (x > 90) {
  document.querySelector(".class").style.marginTop = x + "px";
}
}
</script>

```

### React

here are components such as Navigation bar, we write code in components because by that we can easily change that code according to the specified type ~~and~~, that fast the project done time


### Props

In components we create a component called form component then we write information in this form and that information is called props then we send that or forward that form to other place.

React single webpage application as this needs single refresh only then don't need to refresh again ~~even~~ still the content refreshes automatically




## React necessary extensions

- (1) Thunder client  
open VS code and click this icon  and search thunder client  
This basically used for check the api
- (2) ~~ES7~~ React/Redux/GraphQL/React-Native snippets  
this used for fast writing the code
- (3) Bracket Pair Colorization Taggler  
used for Brackets colorization such as  

{ }
{ }
{ }
{ }

blue color
red color
- (4) Auto Rename Tag  
used for whenever we change the tag name then its ending tag automatically changed
- (5) Prettier - Code formatter

React 90% of the code written in App file  
due to React cause we break this App file  
into number of components such as complete App.js  
for javascript based App file, App.css for css based  
App file

for use CSS under App.js file we do  import that  
file in our App.js file by `import './file name'`  
single quotes
file name  
 By default project accepts src  
 keyword as folder name in App.js  
dot slash  
lays under folder with extension



In Javascript we read that function body must need function call for to do all the tasks

Here function body no need of function call here we use return type keyword for showing javascript code to output

we write return type keyword as part of function body

```
function App() {  
  return ( // return type keyword  
    <div>  
    </div>  
  );  
}  
export default App;
```

html  
this code written for showing at output  
is that html code shows at output  
this to be study further

In CSS we use class keyword for showing that is class with some name

But in react we use className keyword

React components are of two types

- (1) function based components  
such as After import line



We write function body code then it is function based component

(2) Class based component  
such as after import line we write ~~class~~  
~~class~~ class Name then curly braces such as  
App.js

class Name  
If we create a another div under function that's create a error for that create another div and take all code whatever written under function body and error get solved

## JSX

In original javascript we need to write html code as under string ~~code~~

But JSX allows there is no <sup>need</sup> of write html code under string, we can directly write html code ~~code~~

for <sup>run</sup> successful of React project by npm start, you must in that folder which contains files such as node\_modules, src, public

## Javascript in Advance

Variable types or notations or create variable concept notation

- Variable " (1) let keyword  
(2) var keyword  
(3) const



variables such as x, y, z or a, b, c can easily store assigned values and you get output on that variable

but in same way we now modernised the way to write variable such as use keywords such as let, var, const before write the variable

~~But if we use~~ ~~var keyword~~ with assigned value, if you use var keyword with assigned value to that, then you get second value is other shown or overwrite, even vs code don't give you error on this too

<script>

var a = 12;  
var a = 13;      output is 13  
</script> document.write(a);

Variable keyword

let keyword

<script>

let a = 12;

let a = 13;

error you already declared the variable  
document.write(a);

</script>

output is blank page  
or check error at console

So in this let is advanced version of



var

const keyword has specific meaning as per particular usage such as this code is written by specific author and other's so can't change this condition, we use const keyword

<script>

const a = "seena";  
document.write(a);

output & seena

</script>

if we write anything to same variable, the output be broken

<script>

const a = "seena";  
a = 12;  
document.write(a);

output &  
blank space

</script>

so const keyword is used once on one variable and then assigned value to that variable

if you write or assigned different value to that same variable in a new line, output be broken

Right way to use var keyword

var a = 12;

var a = 13;

same variable but two var keyword not, wrong

right way

var a; // create variable by var keyword  
a = 12; // initialize value to that variable



### Const

const can't relate to both of let and var keyword

### var

var a = 12;  
var a = 13;

(1) You not get error of same variable assigning different values

(2) Output is overwrite of a second value assigned

(3) Here redeclare possible and assigning that to same variable possible too such as

~~let~~ a = 12;  
var a = 13;

output is 13  
document.write(a);

var a;  
a = 1;  
a = 2;

output is 2  
document.write(a);

### let

let a = 12;  
let a = 13;

(1) You got error of same variable assigning different values

(2) Output is broken here due to error comes

(3) Here redeclare of same variable not possible but assigning to that same variable is possible such as

let a;  
a = 1;  
a = 2;


document.write  
(a);  
output is 2



## React


Run react project output onto your phone or device, by default where you write code, that system has access to that project by local host ip such as for react vs code local host ip is `localhost:3000`, if you want to watch `http://` this ip on your phone, your phone connected to that wifi which your code written device connected, then by type this ip on your phone and you got that project output.

## Ctrl +

when you want to search anything, use that shortcut but you ~~click~~ click > arrow sign you'll get replace tab in which you have to write replaced keyword such as you search for excel but you want to change that into axis, for that type axis in replace tab and then click this icon  for successful done this change.

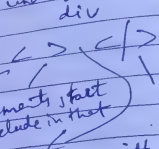
In href don't use # sign that's irrelevant in react, for that you must replace that by / slash sign.

with Ctrl + shortcut type "#" then replace tab keyword "/"

tags with no ending tags ~~use~~ use / sign at the end of the tag or before this arrow > such as  />



In some cases, you have already built or written code that is not under or you don't want to use ~~it~~ too for that use fragment.

Also known as ~~JSX~~ fragment.   
It acts as a frame or all other that code to be written here.

node-module is a dependency of your project if ~~it~~ mistakenly deleted this, then type npm install or npm i and you get that again dependency in your project.

Don't write code into your public folder

App.js return type actually catched at index.html file ~~id~~ id="root"

you want to Now use ~~js~~ bootstrap in alternative to CSS then type bootstrap on google and on introduction page you get CSS code with ~~link~~ href, copy that and paste it into your index.html file under public folder under head tag.

<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" ...>

in  
Advanced JavaScript



for loop based more concept which to array and object

Green House  
Date \_\_\_\_\_  
Page No. \_\_\_\_\_

① `forEach()`

② `for in` (that's not complete syntax) that just ~~is~~ keyword

③ `forEach()` which is basically for array

array syntax  
variable name here = `[0, 1, 2, 3, 4]` ;

`forEach` syntax with array

variable name here `forEach` (here is inbuilt function) ;

that's basically `function()`

In inbuilt function we use formal parameter such as `x, y, z` or `abc` or value, index

A function without name or here we don't need to create other function body or not call that too

<script>

`var x = [1, 2, 3, 4, 5];`

`x.forEach(function(`

formal parameters are `x, y` for getting output

// write here what tasks to do

);

`document.write(x + ":" + y + " <br>");`

output element  
value 2 : 1

here it is index

or be automatically selected for input demand, y for index





~~function body~~ ~~parameters~~ ~~key/value pairs~~  
we can do for each by <sup>create</sup> function body too

x.forEach(~~abc~~); <sup>function call here</sup>  
function abc(<sup>formal parameters here</sup>)  
{  
 document.write(<sup>or x if we don't use () bracket at last time</sup>~~abc~~ + " " + " of " + "<br>");  
}

function body

② <sup>generally</sup> for it, 'used for object'  
object covers key/property and their value

Syntax

variable name here =

key/property here : value  
name = "Rohi"

if you try to ~~get~~ show object as output you'll  
get not nothing such as this text [object object]  
for this use, if you want to see value of  
name key/property  
x.name on document.write(x.name);  
give you value of that key/property



## IMPORTANT NOTES

for in Syntax

for (let "A" in

for

write variable name here

variable name

in

standard keyword

here that must matches your object variable name

for pretty output

{ write code here for what tasks to do }

<script>

var variable name here = { key/property: value }

name: "rohit",  
class: "rohit"

look by

for (let "write variable name here for pretty output" in "here that must matches your object variable name") {

keyword

here is x

This ~~is~~ automatically considered as [y] ~~as a variable~~

document.write (y + ":" + x[y] + "<br>");

output is name: rohi  
class: rohit

as output for key values only

</script>



## IMPORTANT NOTES

~~Variable name taking~~

~~for taking output~~

~~usually given as key with the value~~



~~condition is variable name of taking output~~

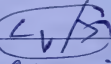
~~must~~ variable name for taking key values only  
must matches the output taking variable


output taking variable automatically accepts as  
key properly







## React notation

-  using for file which is available at same location or path such as by src is set then i created a new folder as default path there, there was already App.js file, so for importing we use  ./ before file name notation

How to include any component in your project such as Navbar, you first to import then use tags 

these tags  for including any component

"write here what to include"  
"here it's Navbar or component name"  
"that's whole a component tag"

src - shortcut used for code fast when you created a blank page then ~~src~~ by src you got already built template according to your file name in which you get import, export, return too, by default you got function based component such as export default function " "   
return (  component name here ( )   
}  this whole got by src

Create a module, module has extension of .mjs linking two modules or get ~~data~~ data to other module by export.



create a filename with name module1.mjs

```
var x = "Rohi";  
var y = "Rohit";
```

create a variable  
here  
and assigning string  
to that

Now

export

for export use export keyword the type default  
and type variable name here to which to export

export with default keyword its a way to export  
or type

export default "variable name here which  
to send" ;  
here its x

Now import this export to your other module  
filename

~~import~~ this

import "write any variable name such from"  
as x, y, z or q, b, c

this not interfere with output  
because its export type

export type

output will be shown on console.log ( )

module location here with  
notation for importing

here its ./module1.mjs



export type 2

```
var x = "Rohi";  
var y = "Rohit";
```

→ create a variables here  
and assigning string to that

Now export

for export use export keyword then here no  
need of default keyword and type variable name  
here which to export

all variables which to  
export must be included in  
that ~~code~~ braces such as  
export {x};

In place of default keyword  
use curly braces

Now import which you already export  
for import do

```
import { "here variable name must matches which  
to export before" } from '  
single quotes'
```

module location here

with ./ notation for importing

here its ./ module.mjs;

output will be shown on console.log("i")

write variable name here in which we do  
importing

here its ./ module.mjs;



Import <sup>and</sup> export is generally a Javascript concept  
 Sending Information through prop/property  
 function ~~the~~ "functionNameHere" ( ) {  
   return (

};

for <sup>first</sup> use of props we <sup>use</sup> this as a ~~parameter~~  
 parameter because may be that function has  
 any task to which we to change  
 such as

function functionNameHere ( ) {  
   return (

    // here is task Navbar ;

);

we want to change that Navbar to Home menu  
 for that

• write <sup>first</sup> props as a parameter to that function  
 function functionNameHere (

    // here parameter is props

  return (

    // here is task Navbar ;

);





Now  
change anything by props keywords  
first write curly braces { }

such as  
{ props-title }

here first write props keyword  
"any genuine keyword here  
(which specifies whole  
line)"  
→ here is title

We do use this keyword for setting anything  
that's means if we do with { props- } genuine keyword  
as hyperlink, we basically can change

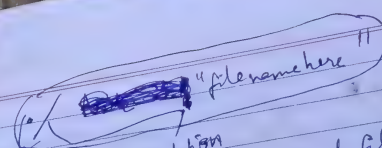
such as  
the hyperlink without moving to that file by  
genuine keyword or if we use that genuine  
keyword in App's file for setting anything  
or assigning that, this will change will replace  
the whole { props-title } is output by assigned  
value

here no need to again write  
props as parameter  
genuine keyword  
import is necessary  
here by  
keyword  
we can  
have or  
write  
component  
is!

such as  
{ > }  
any not important  
function APP { }  
returns {  
number title = "  
any only value  
here  
import Number from  
component  
is!

Node and Tab key combination  
open terminal and type Node and click  
Tab key and write files with ./ notation and location  
to



Node  "filenamehere"  
 gives notation output of that file under terminal  
 such as module: for export  
 module: for import  
 Node ./ "filenamehere"  
 notation module: info

When you creating any component first letter be in Capital such as ~~Navbar~~ Navbar.js

for simplicity we first create components folder then write components ~~to~~ to that

props provide you dynamic message modules also used for send information

propTypes or props  
~~propTypes~~ validation

we first have to use shortcut and you get import propTypes from 'prop-types';

Then syntax  
 " " propTypes =  
 component name here ~~to apply this~~

`{  
 "genuine keyword here": propTypes."  
 };`  
 Keyword  
 value type here such as string, number  
 here its string,





in the case of number

~~dependent on the~~ <sup>here</sup> keyword  
"genuine keyword" = <sup>here</sup> `propType: number,`

in props if you assigning number type value to the genuine keyword then that value must be <sup>for number type value</sup> `number` curly braces `{ }`

< Nav for "genuine keyword" = <sup>here</sup> `<Nav for` <sup>write any number here</sup> `<Nav for`

That means we declaring genuine keyword is of what type ~~such~~ such as number, string

default props  
If you mistakenly not assigned any value to the genuine keyword then by default, default value works

System <sup>import from here</sup> `import { defaultProps } from 'react'`  
"default props" such as  
"genuine keyword" here = `value here`

PropTypes basic usage is to know that genuine keyword assigned value of what type or we can set that



That means we do create a validation here such as when assigned a number to genuine keyword this will give you a error on console such as ~~failed prop type~~ 'Invalid group 'little' as ~~failed prop type~~ 'expected 'number' of type 'string' supplied to 'Navbar', after writing of type string validation for that after writing because we created validation for that after writing a whole component we do create a validation

validation is basically propTypes with a delay value must be of which type so that we get an console if we assigning wrong value

such as under component this first we will create

```
function Name() {
  return
```

```
}
```

export default ~~name~~, then do

~~propTypes~~ propTypes write in newline

condition is propTypes validation must be written under in component in which we using props

• after write whole that component code of

```
function Name() {
```

```
}
```

export default

```
propTypes {
  Report default function Name() {
```

```
}
```





classmate  
Date: \_\_\_\_\_  
Page: \_\_\_\_\_

export default  
at the end  
as this  
will export  
whole function  
~~as this~~  
as whole component  
for use that into  
other file

this will export  
whole function such as  
function App() {  
 return (  
 )  
}

## Javascript in advance

Rest  
if you have 6 elements, 3 be selected for  
3 variables and left one are used for Rest  
on console Rest give your left elements in a  
array form

<script>  
function abc() {  
 // here are two variables generally  
 x, y, "Rest operator"  
 // with new  
 // variable "  
 ...abc  
 document.write(x+y);  
}

function abc(x, y, ...rest);  
// here are two variables generally  
// with new  
// variable "  
// ...abc  
// document.write(x+y);  
// }  
// In most case add more actual parameter  
// than formal parameter  
// left one elements you got to Rest as  
// array  
// for Rest to be  
// working formal parameter has Rest syntax do  
// not as ... "this here any variable name"  
// > abc()



on console, abc1 gives you left elements  
 output is (1,5,6) of rest on  
 console(abc1);  
 0:5  
 1:6

Spread in object

Now to update the object property value

<script>

```
var x = {  
  name: "Rohi",  
  class: "Rohit",
```

here its Rohit

// update by this x.name = "shrey here whom update";

now check on console

```
console.log(x);
```

output is

```
{name: 'Rohit', class: 'Rohit'}
```

value

Now update any property of object by spread

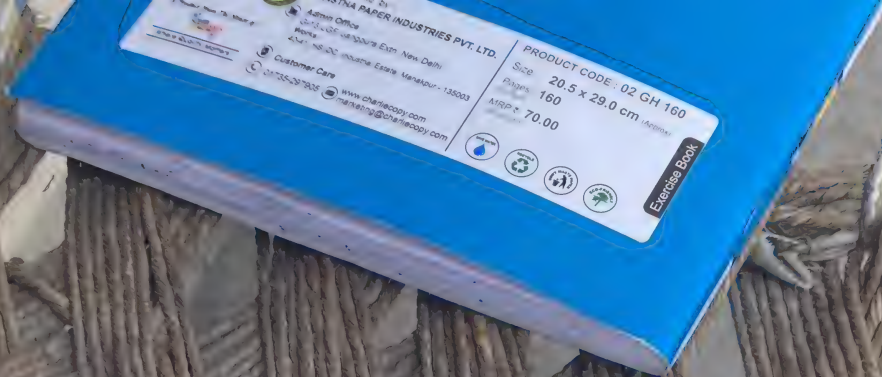
Now create a new variable here  
 new variable // var y = { ...x } curly braces

this is  
new  
variable

variable name to which included  
 or based on to whom you  
 property value to be linked  
 for changed here its  
 x!

spread  
operator





"write here what to update in property value"

↓  
here its class property and its value to be update  
such as class: "Rohit";

↓  
on console check this  
console.log(y);  
outputs {name: 'Rohit', class: 'Rohit'}

Rest in object

block x = {  
 name: "Rohit",  
 class: "Rohit",  
}

New way to show value or get a value  
 curly braces " " here enter variable name from where you get  
 var { } = { } here enter property name to whom you get value such as name

on console log  
console.log({ })  
output is  
Rohit  
here enter property name here to which you entered under curly braces  
here its name  
such as there variable name is x; as object elements  
That was one element, left as where goes  
so we use here Rest

→ So we use ... for using Rest



var <name>, abc  
↓  
Postoperator  
← other variable

on console  
console.log(abc);  
output is  
← class: Rohit

Simple function

function body (function abc())  
↓  
call here // abc();

types: function in modern way

~~function~~ var "function name here" = function()  
↓  
here its abc  
curly braces

keyword

// call here abc();

types var "function name here" = [ ] => <sup>here not function keyword</sup> <sub>at that here</sub>  
curly braces  
↓  
here its abc  
// call here abc();

(That's generally called arrow function)





if function has only one task to do on line line  
 then simplified way of writing function is

var "function name here" = (  $\rightarrow$  "here task to do" )  
 square braces  
 array  $\downarrow$   
 console.log (  $\downarrow$   
 any variable  
 here ) ;

// call here abc() ;

more way to this  
 if you have only one parameter without square braces

var "function name here" = ( "parameter" )  $\rightarrow$   
 parameter here  
 here its x array  
 "here task to do"  
 $\downarrow$   
 console.log ( x ) ;

// call here ~~abc~~ abc (  $\downarrow$   
 actual parameter here ) ;  
 $\downarrow$   
 here its 3

spread in array

var z = [ "Rohi", "Rohit", "Rohiti", "Rohitz", "Rohit3" ] ;

function body {  
 function abc (  $\downarrow$   
 sub body, several parameters here  
 such as x, y, z, a )  
 document.write ( x + y + z + a ) ;  
 }



variable name here which you entered at creating array here its 2

call here  $abc(2)$ ;

This way or  $abc(2)$  without spread outputs ~~contains~~ all ~~variables~~ and that's why  $x, y, z$  and a variables get undefined

To fix this use spread in array

instead of  $abc(2)$  use  $abc(...z)$ ;

spread operator individually substituted as per the formal parameters number of actual parameters and in output undefined got to be not shown

such as  $x$  for 1st element  
 $y$  for 2nd  
 $z$  for 3rd  
 $4$  for 4th

## React

Hook means without creating a class use features  
useState is type of Hook

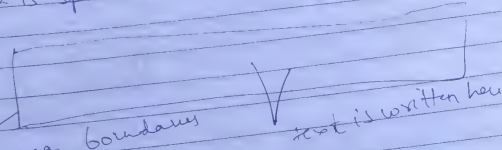




State means situation which right now, after type what situation is

textarea is space where we write any text

that means  
we have to take  
value  
by default  
written  
value



generally by bootstrap we get textarea snippet such as

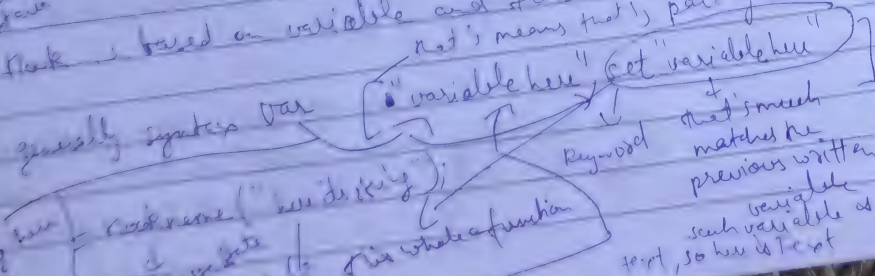
```
<textarea class="form-control" id="exampleFormControlTextarea1" rows="3"></textarea>
```

for this go google and search first bootstrap and in left side find form control, click that and now scroll down and get example tab, under this you get two div and you second one for textarea, use that without label tag or simply use textarea tag

This will show a textarea at output

if you want to increase textarea height use rows="11" that is value as number higher than 3, that's will add more rows to that

Hook is based on variable and function that means that's part of hook



generally syntax var hook name("variable here")

hook name("variable here")

hook name("variable here")



on charge event ~~in a~~ ~~event~~ ~~to~~ ~~be~~ ~~the~~ ~~only~~  
~~when we are trying~~ ~~making~~ ~~linked~~ ~~to~~ ~~that~~ ~~we~~ ~~are~~ ~~being~~  
~~regarded~~ ~~as~~ ~~the~~ ~~primary~~ ~~forces~~ ~~of~~ ~~a~~ ~~parameter~~



such as

value function name = ( ) =>

here its event keyword

onchange is used for when we type anything under empty space that's not directly shown over then we create mechanism to work that such as when we type, type text should be shown on output or under blank area, for that we type from any device that's not directly shown on output, we to update that to be seen on output

must has event parameter or keyword event

so function or setText is a part of onchange event too for onchange event to be work for that function linked to that event. target.value - actually that's getting the value of text, that may be number or string or even character

~~the text value~~ we used this getting value to update the text or that's text comes as parameter for function or setText

setText(event, target.value) - value got here

setText updating not getting value

Direct functions

such as toUpperCase() that's convert given text to the uppercase

toLowerCase() that's convert given text to the lowercase

we used event keyword here to target the value that's means for onchange we first create function which has event parameter then use the same event keyword to target value or get value

what not  
onchange  
event  
not work



Textarea tag can work without CSS too so we can use them and textarea be shown on output

How to make Record of your text or each character

such as  
Total character = 5 - on type 5 character in your  
for Total word  
mechanism is use Total character = {text.length}

length keyword convert each character to count or in number

early for  
browser for  
we accept  
here  
available with  
some property

mechanism use Total words = {text.split(" ")}

split(" ") works as space or the add space when we type space on your device  
split(" ") add space or empty string to that  
such as first write A , without split(" ")  
function space not works  
device, so when you give space between two character A B will give you 2 words  
space and length  
keyword

Preview is just copy of textarea which don't to edit, it's just to watch as textarea text be written, here be text written seen





~~But when I click on the button~~  
~~the color changes~~

do it made if-else

if ("attribute here" == "color here")  
strict equality better than

// update color here  
object property value  
set "attribute here" ==  
color: "white",  
backgroundColor: "black",  
}

// update here button string  
setButton ("string write here to which to show on button");  
curly braces

else  
set "attribute here" {  
color: "black",  
backgroundColor: "white",  
}  
setButton ("string write here to which to show on button");  
curly braces

generally  
here write property name } = "here object"  
to whom value name to whom you  
show at output property name linked



in React  
actually this is opposite  
"have property name" = { "here object name to whom your property name is linked" }

State is used to do multiple works at one time  
such as  
state {  
 update text  
 update object  
 update button string  
 }  
 for update create one state  
 for update text create one state too  
 for update button string create one state too

That's means different update needs different states

Similarly for getting object into attri built  
"attri built here" = { "object name here" }  
value is also an attribute

Now we create a mechanism to ~~copy~~ ~~what ever~~  
~~written under text~~

~~copy~~ ~~direct function~~ ~~function~~ ~~function~~ ~~function~~



~~function name here~~ ~~with the return~~  
~~function name here~~ ~~with the return~~  
~~function name here~~ ~~with the return~~  
~~function name here~~ ~~with the return~~

~~function name here~~ ~~with the return~~  
~~function name here~~ ~~with the return~~  
~~function name here~~ ~~with the return~~  
~~function name here~~ ~~with the return~~

JavaScript in advance

new way to call back function

<script>

var "function name here" = ( ) => {  
such as      setTimeout ( ( ) => {  
abc              ↓  
                  password for  
                  setting the time  
                  for daily  
                  the task  
                  } , "here time in ms");  
                  ↓ as  
                  such as 2000  
                  for 2s);  
                  ↓  
                  for daily task  
                  after 2s

use that same function here  
which previously  
used

So here, no need  
of write again  
function name

// call here      abc  
                  ( )      // would be  
                  <script>      }





if in case this structure you got such as  
`"functionname" ("functionname")`

Part 1 means `functionname` first calls then  
`functionname2`

Direct functions also called task functions such as  
`functionname()` <sup>square bracket</sup>

↓  
 here function name also tells task  
 what task to do

That's means <sup>direct</sup> "function" exists with inbuilt task

Total character = "get here any number"

for that

mechanism is that

use `{ }`

curly braces

~~for~~ to show number as  
 output

mechanism is "variable name" "property name"

↓  
 here its text

↓  
 here its length

~~JavaScript~~ In advance

~~calling~~ calling other function first even the  
 first function time is less

(script)

`var abc = () => {`

`setTimeout(() => {`



```
document.write(a);  
{, 2000};
```

```
var abc = () => {  
  setTimeout(() => {  
    // callback call here  
    {, 2000};  
  }, 2000);  
}
```

for that, any variable to do as parameter to second function, that function acts as formal parameter, so, this formal parameter accepts whole first function as value to parameter such as

```
() => {  
  setTimeout(() => {  
    document.write(a);  
  }, 1000);  
}
```

In that first function not works first even first function. `setTimeout` is less than second function. Now we have to call that function or `xc()` after done our necessary work of second function `xc()`; (really meaning first function calling here)

```
<script>  
var abc = () => {  
  setTimeout(() => {  
    document.write(a);  
  }, 1000);  
}
```

whole function  
not to  
x

```
var abc = () => {  
  // any variable here  
  // here it's x  
}
```

done





now x use for call at time  
set timeout ( ) => <  
a=90;  
x();  
7, 3000);  
}

abc1(abc);  
abc1 first call  
function  
then abc function

asynchronous or async function or async()

async  
async means if we first  
go for first task  
simultaneously we fix  
second task, even third  
task  
here multiple works  
done at same time

sync  
sync means if we first one  
task then go for second task  
here one work at one time

~~async~~ await keyword  
for reducing the overload due to asynchronous  
we use await keyword ~~for add~~ for add  
sequence to your tasks such as this task must done  
first then go for second task  
for that we use await keyword before that task



keyword  
await "task 1 here" // as this task successfully done then  
await "task 2 here" task 2 goes to work

for using async function, you must create other function for use that variable

```
<script>
var "newfunction name here" = async() => {
    // here abc3
}
```

// now call this function here  
abc3();

### Javascript advance

#### Promise function

It's used to return something

we use call for  
get first  
function  
then other

```
<script>
var "function name here" = () => {
    // task here
}
var "newfunction name here" = () => {
    // call here
    // and get() successful then we do use catch()
    // on successful call of set() then must call get()
    // do not use .then to dot set()
    // write here which function to call
    // write any new variable here for get error
    // such as error
}
```

set().then().catch()





classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

Now what we got error in error variable we have to view that to console for that we use arrow function such as

```
catch((err)) => { console.log("same variable here that must makes an previous one") }
```

Promise function accepts two parameters here is error such as except or reject, or we say resolve or reject, resolve uses when not on get any error, reject to get any error

callback function with promise return

```
var abc = () => {  
  return
```

now we ~~use~~ use "new" keyword before writing promise function which to return, that's necessary too so that promise function works such as  
variable = () => {  
 return new Promise ( )  
 // keyword

that has two parameters resolve or reject

```
(resolve, reject) => {  
  setTimeout ( ) => {
```

x = 90;  
Now we working on mechanism that error get then got reject, otherwise resolve so in that case we previously or by default set that variable to be which has stored previous value



for first  
"create variable here"  
was "variable here" = "shared value here"  
such as  
~~any~~ here it's false ;

```

if (!error)
    resolve(function() {
        resolve();
    });
else
    reject("write here what to show");
}, 2000);

```

Generally `export` function is used with `Promise` function.

That's means we do first use Promise function in `then`, then we go for async function. ~~the previous function~~

Use Groups in your project in HTML  
first search google maps or google then click  
three lines = at left side then click share or  
embed map then click embed a map, copy that  
iframe tag link, paste that in your html file  
under body tag





Direct function with number of direct functions  
~~where we use direct~~  
we have to remove selected blue background  
after that successfully selected

for selected we use select();

now we want to remove background so  
we first get selected area for that we  
getSelection() then now these selected background  
by removeAllRanges() remove area or

such as  
document.getSelection().removeAllRanges();

How to count that space is not considered as a  
whole word

generally we do use "variable here".split("").length

But that's counts space as whole word  
for we to work before length or total word count  
now it's only way to overcome this to use new function  
here called filter function and to apply condition  
to that too

such as

filter(v) => { return " " + variable here  
" write any variable here  
such as element  
which you used  
before  
such as element  
length != 0 }  
properly

That's means we do condition here if any variable here  
called element length not equals to zero then



A new function  
 Squared braces  $\Rightarrow$   
 Not a whole  
 a new  
function





classmate  
Date \_\_\_\_\_  
Page \_\_\_\_\_

Arrow function is of ~~two~~ types  
① Synchronous function ② Asynchronous function

~~Asynchronous function works with~~ ~~asynchronous function~~  
Promise function, setTimeout function, not  
works without async function or it's part  
output not shows

Arrow function is used to create one function  
under other function  
such as

```
var abc = ( ) => {  
  setTimeout(() => {  
    // "function in a function"  
  })  
}
```

that's a  
arrow  
function too

for your project easily be searchable by the  
Search Engine Optimization  
Go under index.html file and go to content  
under head tag and write keywords under  
" ", such as Text converter, convert text  
uppercase to lower / lowercase to uppercase  
not's or

Function Based Components  
(1) Stateless  
(2) New Props are directly used  
such as { props: "keyword" }  
(3) After updation these able to  
use usestate hook

Class Based Components  
(1) Statefull



Props are immutable or not changeable  
such as

< "component name here" "keyword" = "its value here" />  
you then again assign value to that same keyword  
< "component name here" "keyword" = "its new value" />

that will broke the output, nothing will display as output

Class based components syntax

first we do import React from 'react';  
syntax class "classname here" extends React.Component {  
 render() {  
 // defines function that runs in class before return  
 return (  
 // JSX goes inside here  
 );  
 }  
}

↓  
keyword keyword  
↓  
this is mainly class name  
↓  
curly braces

Props in class components

in function we used here to use  
< this.props "keyword" />





Proops with children

for that component included must be  
turn into ~~tag~~ tag such as  
<Class /> - A whole component tag  
that will turn into

<Class> </Class>

Now create children to this tag

<Class "Keyword here" = "value here">

// children write here

<h1> Hello </h1>

<h2> Hello! </h2>

</class>

we don't need to write again keyword

so that means that children linked to that keyword  
so we first create proops for that keyword then  
for that children

such as

<h1> Rohit <proops "keyword here" /> </h1>  
<proops children />